Alma Mater Studiorum · Università di Bologna

DOTTORATO DI RICERCA IN MATEMATICA
XXII CICLO
MAT 07: Fisica Matematica

ENTROPY AND SEMANTICS: TEXTUAL INFORMATION EXTRACTION THROUGH STATISTICAL METHODS

Tesi presentata da Chiara Basile

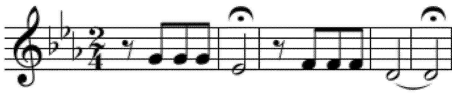
Coordinatore:
Chiar.mo Prof.
ALBERTO PARMEGGIANI

Relatore:
Chiar.mo Prof.
MIRKO DEGLI ESPOSTI

Esame Finale anno 2010

Introduction

There are more things in heaven and earth, Horatio,
than are dreamt of in your philosophy.
- W. Shakespeare, *Hamlet*, act 1, scene 5 -



- L. van Beethoven, *Symphony no. 5*, Allegro con brio -

GACTGCTTGCATTAAAGGACTTCCTCATC...
- Human DNA, *chromosome X*, gene ODZ1 -

When William Shakespeare wrote his masterpieces, he certainly didn't think that, one day, someone would refer to them as *productions*; for sure, while typing on his piano keybord, Ludwig van Beethoven didn't consider himself as an *information source*; and if we think of the sequence of bases in a gene, our first idea will not be to look for *patterns* so as to be able to give a shorter description of its structure.

The relationship between information theory and the real world is not immediate, and the risk of confusing the reader is always present. Nevertheless, a large number of situations in our lives fit into this scheme: someone or something "produces" an output, that can be represented as a sequence of symbols extracted from some finite set; even if we do not know the details of the source, we can reconstruct its properties by looking at the sequence it generated, by extracting as much information as it is possible to deduce

from its content, structure, etc.

Let us go back to Shakespeare's example: we have few details about his biography, and nevertheless when we read *Macbeth* or *Hamlet* we acquire a lot of information about his beliefs, his wit, his idiosyncrasies, his use of words, etc. All of these elements can be included in the single, comprehensive concept of *style*. And if the normal path for the detection of an author's style is the study of his bioghaphical and historical background, together with an accurate (human!) reading of his writings, there are a number of properties that can be detected by using automatic or semi-automatic tools. This is exactly the object of research on *automatic information extraction from symbolic sequences*, a very wide and lively field of information theory, with a quite solid mathematical background and a number of contaminations with other fields of research, as the theory of dynamical systems.

Since most of the information in the world can be represented in terms of symbolic sequences (cf. [13]), the methods and models of information extraction can be applied to a number of cases. Among others, and with no pretension of completeness, we add to the already cited problem of authorship attribution a large variety of text classification tasks (by genre, by topic, ...), the automatic extraction of keywords from long compositions [49, 50] and, stepping away from written texts, the identification of musical pieces from short and noisy excerpts, a number of biological sequence classification problems [8, 20, 52], and the list could be much longer.

Some of the techniques that are used in literature for these and other similar tasks exploit as much as possible the particular structure of the sequences that are the object of analysis: for example, a written text has an underlying syntactical structure that can be put in evidence by an appropriate tagging of words or phrases, whereas a piece of music can be studied in terms of its sections, beats, pauses, and so on. The methods that we will apply for information extraction from written texts, instead, will be as much as possible *independent* from the structure of the texts themselves, and will refer to very general theoretical results in terms of entropy estimation

and statistical properties of the sequence; this is a tendency that is gaining ground in this field, usually through the studies of researchers coming from "pure" sciences like mathematics or physics.

An important remark is that often the study of experimental results leads to interesting considerations about the theoretical framework that underlies the methods, and in our experience the practice of information extraction often requires the use of *semi*-statistical indicators: asymptotic properties and statistically significant quantities are not often the case in real-world textual corpora, and the study of this "borderline" region may lead to interesting progresses both for theory and for the applications.

Here is the scheme of this thesis.

In **chapter 1** we will present the theoretical framework of symbolic sequence analysis, adopting mainly the language of Information Theory, as founded by C.E. Shannon in [59], but without forgetting the equivalent formulation in terms of Dynamical Systems; a very stimulating reading to understand this equivalence is the book by P.C. Shields [60], that was the source of inspiration for a good part of chapter 1. After presenting the definition of *information source* and discussing the meaning and the central role of properties like *stationarity* and *ergodicity*, also for the applications, we will move to *data compression* and its role in the estimation of the *entropy* of a source. Data compression is nowadays a very well established field of information theory, thanks to the founding papers published by J. Ziv, A. Lempel and their coworkers in the 1970s (cf., among others, [38, 70, 71] and the review paper [69]), where they proposed a variety of compression algorithms (the family of *LZ algorithms*), based on the idea of a clever *parsing* (subdivision) of the symbolic sequence. It was a huge progress in the field, since it was the first example of compressor that doesn't operate a fixed number of character at a time, but is allowed to vary the length of encoded substrings according to the "size" of the regularities that that specific sequence presents; indeed, such algorithms are still at the base of the most common zipping software that we use everyday on our computers.

In 1993 J. Ziv and N. Merhav [73] proposed a method to estimate the relative entropy (or Kullback-Leibler divergence) between couples of information sources, starting from their realizations: they proved that a modified version of an LZ algorithm, where the regularities for a sequence are searched in another sequence, can be used to approximate the relative entropy between the two sources that generated such sequences. This important result was used in various subsequent studies, among which [8], to deal with problems of text classification and clustering.

Taking advantage of some results on the recurrence of patterns in symbolic strings, that were developed in parallel in the context of dynamical systems (in terms of hitting times for the orbits of dynamical systems, see for example [21]) and with the language of probability and stochastic processes (cf. [33, 34]), we will give a new proof of Ziv-Merhav's Theorem. Our proof is different from the original one, which is based on heavy probabilistic tools, and also from the one that H. Cai, S.R. Kulkarni and S. Verdú proposed in [11], where they make use of the Burrows-Wheeler transform [10].

With **chapter 2** we will move from a purely theoretical setup to the applications to information extraction from written texts. In this chapter we will deal in particular with problems of *authorship attribution*, using methods that are derived from the results in chapter 1. With this we do not mean that we will *directly* apply theoretical results of information theory to real-world contexts: as discussed above, indeed, the passage from theory to practice always requires great care and some adaptations to fit the specific problem.

We will describe in this section two experiments with real-world textual corpora. The first is the result of a collaboration between our group, D. Benedetto and E. Caglioti from Rome-*La Sapienza*, M. Lana from the University of Western Piedmont in Vercelli and the *Fondazione Istituto Gramsci* in Rome. We were asked to give indications about the possibility that a number of short articles, published anonimously on the same newspapers where Antonio Gramsci and his coworkes wrote, are actually pieces of Gramsci's writing; the *Fondazione* is passing the results of our study to his team of

expert phylologists, so as to decide whether or not to include those texts in the complete edition of Gramscian works that is in the process towards publication. Apart for the interest of the attribution problem *per se*, the corpus that was made available to us is a very stimulating one from the point of view of authorship attribution studies, as we will thoroughly discuss in the chapter.

Furthermore, it is interesting to note that our studies, that certainly are out of the mainstream of standard linguistic analysis of texts for authorship attribution purpose, are nonetheless consistent with a tendency that started gaining ground in the last decade or so in this area (cf. [32, 14, 29, 61, 62] and the discussion in [6, in Italian]): the written text, which is the result of a complex combination of semantic, linguistic and historical properties, is deprived of all such structure and considered instead as a mere sequence of symbols from a certain alphabet, on which statistical and semi-statistical methods can be applied to extract stylistic information.

The same tendency has become prominent also in the field of *plagiarism detection*, that will be the subject of **chapter 3**. We move here to a more "technological" application, at the same time very different and sharing similar elements with authorship attribution. Indeed, from a certain point of view the problem of recognizing the authenticity of a written text is certainly a matter of style analysis, and in this it is near to authorship attribution; on the other hand, plagiarism is much more content-related than authorship is (cf. [16]), and also the typical size of a reference corpus for this kind of studies is orders of magnitude larger than a standard real-world authorship attribution data set.

In 2009 the *1st International Competition on Plagiarism Detection* [54] was proposed, in order to compare plagiarism recognition methods and technologies on a common ground. We participated in the contest (cf. [4]), and obtained the third position among the ten participating groups: a good result considering that our background on plagiarism detection was practically null. Indeed, we applied there our experience on authorship attribution, es-

pecially for a first selection of the most relevant sources of plagiarism for each suspicious text, which is the first phase of practically any plagiarism detection algorithm (cf. [64]). We also developed a couple of interesting ideas to reduce the computational load of the experiments, by using certain lossy codings that allowed for a reduction of the alphabet and, in one case, for a dratic cut of the text length. One of these codings, based on word lengths, is analysed in a certain detail in this chapter, and with greater accuracy in a work with A. Barrón-Cedeño and P. Rosso of the Polytechnic University of Valencia [2]. As we will underline in the conclusions of this chapter, more work needs to be done to reduce the heuristics and to give our methods a better experimental validation in this field.

Most of the numerical experiments for this thesis were performed by using *Mathematica*© 7.

Contents

Chapter 1

Entropy, relative entropy and pattern matching

My greatest concern was what to call it. I thought of calling it 'information', but the word was overly used, so I decided to call it 'uncertainty'. When I discussed it with John von Neumann, he had a better idea. Von Neumann told me, 'You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name, so it already has a name. In the second place, and more important, no one really knows what entropy really is, so in a debate you will always have the advantage.'

- C.E. Shannon in a private conversation, according to [66] -

1.1 Notations and main concepts

1.1.1 Basic notations and definitions

Let $\mathcal{A}$ be a finite set of symbols, which we will call *alphabet*. $\mathcal{A}$ can be any finite set, from the simplest possible binary alphabet $\mathcal{A} = \{0, 1\}$ up to those of written human languages, containing tens of symbols (letters, punctuation marks, etc.). We will denote by $\mathcal{A}^* = \cup_{n\in\mathbb{N}}\mathcal{A}^n$ and $\mathcal{A}^{\mathbb{N}}$ the sets of, respectively, finite and infinite sequences of symbols from $\mathcal{A}$. In the following $\mathbb{N} = \{1, 2, 3, \ldots\}$, the set of positive integers.

Now that we know the alphabet in which our symbolic sequences will be written, we are interested in giving a model of the entity or mechanism that generates them. The *information source* is represented differently in the different fields of mathematics and computer science that deal with symbolic sequence analysis. Mathematical physicists will more naturally consider the symbolic sequences to be generated by a *dynamical system*, where at each iteration a new symbol is added, corresponding to the element of a partition of the phase space that the orbit intercepts at that time step. In the context of Information Theory it is more common to use a probabilistic representation, in terms of *stochastic processes*.

Definition 1.1. An *information source* is a stationary, ergodic stochastic process.

The two definitions are exactly equivalent, and can be transformed one into the other by adopting the right perspective: see for example [60].

We need here a few basic definitions, that will give us also the possibility of fixing the notations for the whole of this work.

Definition 1.2. Let $(\Omega, \mathcal{S}, \mathbb{P})$ be a probability space and $\mathcal{A}$ a finite alphabet. A *stochastic process* is an infinite sequence $X := \{X_n\} = \{X_1, X_2, \ldots, X_n, \ldots\}$ of random variables $X_n : \Omega \to \mathcal{A}$. The process is *stationary* iff $\mathbb{P}(X_1 = a_1, \ldots, X_n = a_n) = \mathbb{P}(X_{1+k} = a_1, \ldots, X_{n+k} = a_n)\ \forall\ a_1, \ldots, a_n \in \mathcal{A},\ \forall\ k, n \in \mathbb{N}$.

For sake of simplicty, we will often denote the sequence $a_1, \ldots, a_n$ with a_1^n, and in the same way X_1^n will be the first n variables of the stochastic process. Stationarity is a natural but demanding request for a source: intuitively, it means that the source doesn't change its way of generating sequences with the passing of time. This is a far from obvious request for a true source that one can find in the applications, as we will see in later chapters; anyway, it is an important property that we will always require from a source.

Since, indeed, in the following we will always deal with stationary processes, we can introduce the following convenient notation: $P(a_1^n)$ will be

used in place of $\mathbb{P}(X_1 = a_1, \ldots, X_n = a_n)$. Indeed, stationarity precisely implies that these joint probabilities, which will be called in the whole the *distribution* P of the process, are invariant under translation. Such distribution is defined on cylinders of the form $[x_1^n] := \{z \in \mathcal{A}^{\mathbb{N}} \mid z_1^n = x_1^n\}$ as $P(x_1^n) := P([x_1^n]) := \mathbb{P}(X_1^n = x_1^n)$. The marginals of P obtained by considering just the first n variables of X is denoted by P_n when needed, otherwise simply by P. The distribution P is a fundamental quantity for a stochastic process, because it determines univocally the process itself: in other words, the particular choice of $(\Omega, \mathcal{S})$ is not particularly relevant, as long as the distribution $P_n, n \in \mathbb{N}$ is given. Note, furthermore, that even the choice of the alphabet $\mathcal{A}$ is not important: the only quantity that matters is its cardinality $|\mathcal{A}|$.

Example 1.1. In general the value of the i-th variable of a stationary process can depend on all of the X_j with $j < i$; if this dependency is limited to a finite number of preceeding steps, X is a *Markov process*. More rigorously, a Markov process is a stochastic process for which the following holds:

$$\mathbb{P}(X_n = a_n \mid X_1^{n-1} = a_1^{n-1}) = \mathbb{P}(X_n = a_n \mid X_{n-k}^{n-1} = a_{n-k}^{n-1})$$

for some $k \in \mathbb{N}$. The process is said to have *memory* k: indeed, the value of the random variable at a given time step depends only on its value in the latest k time steps, while it is independent of what has happened before.

When $k = 1$ the process is also called *Markov chain*. In this case, by stationarity, the *transition probabilities* $p_{a_1 a_2} := \mathbb{P}(X_2 = a_2 \mid X_1 = a_1)$ define the process completely, and they can be represented in a *transition matrix* $P_{ij} := p_{a_i a_j}, \forall\, a_i, a_j \in \mathcal{A}$.

Note that $k = 0$ means that the random variables composing the process are independent of one another, and in this case the stationarity of the process is equivalent to its being an independent identically distributed (i.i.d.) process.

What does it mean for a stationary process to be ergodic? As usual, the answer depends strongly on the quantities that are interesting for the

problem into consideration. A possible definition states that a dynamical system is ergodic iff no set of positive measure (except the whole space) is invariant under the map of the system.

For the purposes of this thesis, though, the most useful definition of ergodicity is the following.

Definition 1.3. Let a_1^n, b_1^k be two sequences in $\mathcal{A}^*$, with $n \geq k$, and $f_{a_1^n}(b_1^k)$ the relative frequency of b_1^k in a_1^n. Let us define the set of *typical sequences* of the stationary process X:

$$\mathcal{T}(X) = \{a \in \mathcal{A}^{\mathbb{N}} | \, \forall\, k \in \mathbb{N},\ \forall\, b_1^k \in \mathcal{A}^k,\ f_{a_1^n}(b_1^k) \xrightarrow[n\to\infty]{} \mathbb{P}(X_1^k = b_1^k)\}.$$

A process (or, which is the same, its distribution P) is *ergodic* iff $P(\mathcal{T}(X)) = 1$, i.e., iff almost every sequence generated by the source itself is typical.

Intuitively, therefore, ergodicity means that almost every sequence that is generated by the process has the same statistical properties. This property is fundamental for the results in this thesis, and it has an interest for applications too: since (almost) every sequence is asymptotically a "good representative" of the whole source, we are somehow justified in using a few "long enough" strings to reconstruct the distribution of the originating source, which is usually unknown. This is more than a vague idea, as we will see in Theorem 1.1.1.

Example 1.2. Ergodicity has an interesting form for Markov chains. Indeed, a chain (or its transition matrix P, which is the same) is said to be *irreducible* if for any pair $a_i, a_j \in \mathcal{A}$ there is a sequence $i_0 = i, i_1, \ldots, i_n = j$ of indices such that all of the transitions are possible, i.e., $P_{i_m i_{m+1}} > 0$ for all $m = 0, \ldots, n-1$; intuitively, this means that the chain can generate sequences that start and end with any two characters of the alphabet $\mathcal{A}$.

It is not difficult to see that if P is irreducible there is a unique probability vector π (the *equilibrium state*) such that $\pi P = P$, i.e., the chain that starts with distribution π is a stationary process.

Moreover, it is possible to prove (cf. [60]) that a stationary Markov chain is ergodic iff its transition matrix P is irreducible.

1.1.2 Entropy and relative entropy

The definition of *entropy* of a random variable X was originally given by C.E. Shannon in [59]. If X takes values in $\mathcal{A} = \{a_1, \ldots, a_k\}$ with probabilities $p_i := \mathbb{P}(X = a_i)$, its entropy is the expected value of minus the logarithm of the probability of X, or:

$$H(X) := -\sum_{i=1}^{k} p_i \log p_i,$$

where the term of the sum for i is assumed to be zero when $p_i = 0$.

A complete discussion on the properties of entropy for a random variable can be found in [17]; we will now quickly move to the entropy of a process. Let now X be a stochastic process with distribution P.

Definition 1.4. The *entropy* (or *entropy rate*) h of the process X is defined as

$$h(X) := \limsup_{n\to\infty} \frac{H(X_1^n)}{n}, \tag{1.1}$$

with

$$H(X_1^n) := H(P_n) = -\sum_{x_1^n \in \mathcal{A}^n} P(x_1^n) \log P(x_1^n),$$

the joint entropy of the random variables $X_1, \ldots, X_n$, which is also known as the $n-$*block entropy* of the process X.

Eq. (1.1) is not the only possible definition for the entropy of a process; indeed, the following is equivalent to (1.1), as it is easily proved (cf. [17]):

Definition 1.5. Let $h_n(X) := H(X_n|X_1^{n-1}) := H(X_1^n) - H(X_1^{n-1})$, the *conditional entropy* of X_n given X_1^{n-1}. The *entropy* of the process X is

$$h(X) := \limsup_{n\to\infty} h_n(X). \tag{1.2}$$

Note that the sequence of conditional entropies $h_n(X)$ is nonincreasing in n for a stationary process: indeed, it is a general fact that conditioning on more variables reduces entropy: $H(X|Y,Z) \le H(X|Y)$. Therefore:

$$h_{n+1}(X) = H(X_{n+1}|X_1^n) \le H(X_{n+1}|X_2^n) = \text{ (by stationarity) } H(X_n|X_1^{n-1}) = h_n(X).$$

Since $h_n(X) \ge 0$, for a stationary process we can substitute the superior limit in (1.2) with a limit. The same holds for (1.1), but there we need to use a result on subadditivity (cf. [60]).

We will sometimes use the notation $h(P)$ instead of $h(X)$, since the process is defined by its distribution.

As for a single variable, also the entropy of a stochastic process is a measure of the randomness of the process itself. Let us give a couple of simple examples.

Example 1.3. Let X be a *i.i.d. process*, i.e. a process where all of the variables X_j have the same distribution $p_i = \mathbb{P}(X_j = a_i),\ i = 1, \dots, |\mathcal{A}|$. Then the joint entropy of the first n variables of the process is simply n times the entropy of any of the X_j, and

$$h(X) = \lim_{n\to\infty} \frac{nH(X_j)}{n} = H(X_j),$$

i.e., the entropy of the process is equal to the entropy of any of its variables.

Example 1.4. Let X be a stationary Markov chain (memory $k = 1$). The Markov property ensures that $h_n(X) = H(X_n|X_1^{n-1}) = H(X_n|X_{n-1})$, and the definition in (1.2) becomes $h(X) = \lim_{n\to\infty} H(X_n|X_{n-1})$. But the stationarity of the process then allows us to substitute $H(X_n|X_{n-1})$ with $H(X_2|X_1)$, so that the limit is not needed and the entropy of the chain is simply the entropy of its second variable conditioned to the first one: $h(X) = h_1(X) = H(X_2|X_1)$.

In an analogous way it can be proven that a stationary process X is Markov with memory k if and only if $h(X) = h_k(X)$, i.e., in this case the nondecreasing sequence $h_n(X)$ reaches its limit for $n = k$.

In the general case of an unknown stationary process no such reduction is possible, and we are forced to find approximations of the process entropy. The following result will be of great importance in this context.

Theorem 1.1.1. *Let X be an ergodic source on the finite alphabet $\mathcal{A}$. Then, for almost all $a \in \mathcal{A}^{\mathbb{N}}$:*

$$\lim_{n\to\infty} -\frac{1}{n}\log P(a_1^n) = h(X).$$

Theorem 1.1.1 is referred to as *Shannon-McMillan-Breiman Theorem*, from the names of those who proved it in more and more general cases; the (weaker) version with convergence in probabilty is also often called *Asymptotic Equipartition Property* (AEP) and sometimes this denomination is extended, with an abuse, to the stronger version. A complete proof of this result can be found in [60, pp. 51-55].

Consider now two independent information sources X and Y with the same finite alphabet $\mathcal{A}$ and distributions P and Q respectively. The *relative entropy* (or *Kullback-Leibler divergence*) between P and Q is defined as

$$d(P\|Q) := \limsup_{n\to\infty} \frac{D(P_n\|Q_n)}{n},$$

where

$$D(P_n\|Q_n) := \sum_{x_1^n\in\mathcal{A}^n} P(x_1^n)\log\frac{P(x_1^n)}{Q(x_1^n)}. \tag{1.3}$$

Note that the quantity in (1.3) is well defined and is finite if $P_n \ll Q_n$ eventually, i.e., for $n \gg 1$ and for all measurable subset $B \in \mathcal{A}^n$, $Q_n(B) = 0 \Rightarrow P_n(B) = 0$. This ensures that a sequence which is a possible production of X (i.e. has non-zero P-measure) is also a possible production of Y (i.e. has non-zero Q-measure). As a convention, the term of the sum for a sequence x_1^n is assumed to be zero when $P(x_1^n) = Q(x_1^n) = 0$.

The relative entropy is a measure of the statistical difference (divergence) between two distributions, and it has the following property:

Theorem 1.1.2. *For any couple of probability distributions P and Q for which $d(P\|Q)$ is defined, $d(P\|Q) \geq 0$ and the equality holds iff $P = Q$.*

Thanks to this result, a symmetrized version of the relative entropy is sometimes used in the applications as a pseudo-distance between information sources; even if it is not a true distance, since d doesn't satisfy the triangle inequality, we will see in the following that this approach gives good results in the applications.

1.2 Coding and LZ compression

1.2.1 Coding and entropy

As we have seen, the entropy of a process is a measure of "how interesting" are the sequences that process produces, of how unpredictable they are. The direct calculus of entropy using either the definition (1.1) or Theorem 1.1.1, though, is not possible in practice, since extracting the statistics of subsequences of growing length would mean exponentially growing the length of the observed string itself, with obvious computational problems, and also problems of availability of such long sequences, if we are dealing with a real situation.

The theory and practice of data compression gives an alternative method of estimating the entropy of a system. What is a *data compressor*?

Definition 1.6. A *code* on the alphabet $\mathcal{A}$ is a function $\mathcal{C} : \mathcal{A} \to \mathcal{B}^*$, where $\mathcal{B}^*$ is the set of finite sequences on the alphabet $\mathcal{B}$. We will often consider *binary codes*, where $\mathcal{B}$ is the binary alphabet $\{0, 1\}$.

A code is said to be:

- *non-singular* if it is injective;
- *universally decodable* (UD) if any sequence $b \in \mathcal{B}^*$ can be univocally interpreted as a sequence of images of characters from $\mathcal{A}$ through $\mathcal{C}$. Note that this is not implied by the non-singularity: for example, the code

$$\mathtt{a} \overset{\mathcal{C}}{\mapsto} \mathtt{0}\ ;\ \mathtt{b} \overset{\mathcal{C}}{\mapsto} \mathtt{1}\ ;\ \mathtt{c} \overset{\mathcal{C}}{\mapsto} \mathtt{01}$$

is injective but not UD, since the sequence `01` of $\mathcal{B}^*$ can correspond to either `c` or `ab`;

- *prefix-free* or *istantaneous* if no code word $w \in \mathcal{C}(\mathcal{A})$ is the prefix of another code word. This implies that this UD code can be reversed with one single reading of the encoded sequence, since during the reading there is no ambiguity with respect to where the code words end. Prefix-free (or, simply, *prefix*) codes are the most interesting for the applications, because of the time saving they allow in the decoding process.

All of these definitions apply to any kind of code. Moving now to data compressors in particular, the most interesting quantity is the average length of an encoded sequence. Let X be a random variable with values in the alphabet $\mathcal{A}$ and distribution $p_i = \mathbb{P}(X = a_i)$, and let $\mathcal{L}_\mathcal{C}$ be the *length function* for code $\mathcal{C}$, i.e. $\mathcal{L}_\mathcal{C}(a_i) = |\mathcal{C}(a_i)|$, where $|w|$ is the length of the word w.

The *average code length* is by definition

$$\mathbb{E}_X(\mathcal{L}_\mathcal{C}) = \sum_{a_i \in \mathcal{A}} p_i \mathcal{L}_\mathcal{C}(a_i).$$

Note that the definition doesn't depend on the values of the code function $\mathcal{C}$, but only on the length of the code words, i.e., on $\mathcal{L}_\mathcal{C}$.

The following result defines a limit for the average code length and relates data compression to (single variable) entropy.

Theorem 1.2.1. i) *For every UD code* $\mathcal{C}$

$$\mathbb{E}_X(\mathcal{L}_\mathcal{C}) \geq H(X)$$

and the equality holds iff $|\mathcal{C}(a_i)| = -\log p_i$.

ii) *There is a prefix code such that*

$$\mathbb{E}_X(\mathcal{L}_\mathcal{C}) \leq H(X) + 1$$

For a proof of this theorem see [17]. Since $H(X)$ is defined as $\mathbb{E}_X(-\log P(X))$, it can be interpreted as the average length of a code that uses exactly $-\log p_i$ characters to code a_i; theorem 1.2.1 then ensures that this choice is the best possible for a UD code.

1.2.2 Universal codes, LZ compressors

Coding functions, according to the definition we gave in the previous section, take *one single* symbol of the string to code and map it to some string in another alphabet (which we will consider to be binary in the following). Clearly, this is not the only way of coding a string: one could assign code words to sequences of 2, 3, 4... n symbols in the alphabet $\mathcal{A}$ ($n-grams$):

Definition 1.7. An *n-code* is a function

$$\mathcal{C}_n : \mathcal{A}^n \to \mathcal{B}^*.$$

The same definitions for non-singular, UD and prefix codes hold here. Theorem 1.2.1 becomes in this context:

i) for every UD n-code $\mathcal{C}$

$$\frac{1}{n}\mathbb{E}_{X_1^n}(\mathcal{L}_{\mathcal{C}_n}) \geq \frac{H(X_1^n)}{n}; \tag{1.4}$$

ii) there is a prefix $n-$code such that

$$\frac{1}{n}\mathbb{E}_{X_1^n}(\mathcal{L}_{\mathcal{C}_n}) \leq \frac{H(X_1^n)}{n} + \frac{1}{n}. \tag{1.5}$$

We can now consider a *prefix code sequence*, i.e. a sequence $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ of prefix n-codes, and define its *compression rate* as

$$\mathcal{R}(\{C_n\}) := \limsup_{n\to\infty} \frac{\mathbb{E}(\mathcal{L}_{\mathcal{C}_n})}{n}.$$

The results in (1.4) and (1.5) then lead to the following theorem (cf. [60]).

Theorem 1.2.2. *Let X be a stationary process with entropy h. Then:*

i) *there is a prefix code sequence* $\{\mathcal{C}_n\}$ *such that* $\mathcal{R}(\{C_n\}) \leq h$*;*

ii) *there is no prefix code sequence* $\{\mathcal{C}_n\}$ *such that* $\mathcal{R}(\{C_n\}) < h$*;*

According to the theorem, h is a tight lower bound for the compression rate of any data compressor, and we are also sure that, for a given process, there will be a prefix code sequence that compresses to the value of entropy of that process.

As we already discussed (and will see in further chapters), though, the source distribution is almost always unknown; that's why we have to introduce a much stronger concept.

Definition 1.8. A $n-$code sequence $\{\mathcal{C}_n\}$ is *universally asymptotically optimal* (or simply *universal*) iff

$$\limsup_{n\to\infty} \frac{\mathcal{L}_{\mathcal{C}_n}(a_1^n)}{n} \leq h(X) \text{ almost surely}$$

for every ergodic process X.

Note that, first of all, we move from an average to an almost sure condition on the realizations of the source and, furthermore, we require optimality of the $n-$code sequence for *all* ergodic stationary sources. A universal compressor is therefore a very interesting object from our point of view: even without knowing anything of the source, if not its ergodicity, we can be certain that the compression rate will go to the source entropy in the limit for infinite sequences. The compressor can then be used to *investigate* the source, to find an approximation of its entropy, which is usually unknown.

Luckily, a result analogue to theorems 1.2.1 (cf. [60]) ensures that such universal compressors exist, and can be built using only istantaneous codes.

Theorem 1.2.3. i) *There is a universal prefix code sequence.*

ii) *For any sequence* $\{\mathcal{C}_n\}$ *of non-singular* $n-$*codes and any process* X*,*

$$\liminf_{n\to\infty} \frac{\mathcal{L}_{\mathcal{C}_n}(a_1^n)}{n} \geq H(X).$$

For a constructive proof of this result see [60, pp. 122-129]. We will now give an example of a universal code, which will have a great importance for the following of this thesis.

Example 1.5 (LZ algorithms)**.** J. Ziv and A. Lempel, together with various coworkers, proposed in the last decades of the XX century a good number of compression algorithms. The key idea behind all of their algorithms is the concept of *parsing* the sequence, i.e. to split it up into pieces in a clever way, so that this separation can then be used to produce a shorter, equivalent version of the string itself. All of these compressors are istantaneous: only a single reading of the compressed sequence is needed for the decoding process. Furthermore, with respect to other very popular algorithms like Huffman coding [26], LZ compressors have the advantage to be *sequential*, i.e., they do not need more than one reading of the sequence to compress: the compression is performed while the sequence is read for the first time.

We will first describe here the version of the algorithm that is presented in [71]. A parsing into blocks (often referred to as *words*) of variable length is performed according to the following rule: *the next word is the shortest word that hasn't been previously seen in the parse*. Every new parsed word is added to a *dictionary*, which can then be used for reference to proceed in the parsing. As an example, let us consider the following sequence:

$$a_1^n = \texttt{accbbabcbcbbabbcbcabbb}$$

The first word will be simply the first `a`, since we have not parsed anything yet. Also the `c` in position 2 will be parsed on its own, but then the second `c` is a repeated word, so that we can go further and parse `cb`, which is a new word. The following `b` is again a new word, then we can parse an `ab` and so on. The final result of the parse is:

`a|c|cb|b|ab|cbc|bb|abb|cbca|bbb`

While it parses the string, the algorithm also does the coding: since we are certain that the prefix of each word that excludes only the last character is already in the dictionary, we can code each parsed sequence simply with:

1. c_n pointers to the positions of the prefix of each parsed word in the dictionary, which cost at most $c_n \log c_n$ bits, with n the length of the sequence to code and c_n the cardinality of the resulting dictionary (which obviously depends on the string itself);

2. a binary encoding of the ending character, the only new one, for each of the c_n parsed words. This will require at most $c_n \log |\mathcal{A}|$ bits, $\mathcal{A}$ being the original alphabet of the sequence to code.

When we calculate the rate, dividing the expected code length by the length n of the sequence, the term $c_n \log |\mathcal{A}|$ goes to zero in the limit and the dominating term is $c_n \log c_n$. The universality of this algorithm is proved in [71].

Lempel and Ziv proposed a number of variants of their algorithms. The one that we will refer to in the following, LZ77 [70], differs from the one described above because the parsed words can be chosen not only in the vocabulary of previously parsed words, but in the set of all the subwords of a certain string that plays the role of a *database*. In the different versions of this algorithm, the database can either be a previously generated sequence coming from the same source, or the part of the sequence that has already been parsed (transient effects disappear in the limit for infinite length).

In this second case, the sequence above is parsed as follows:

`a|c|cb|ba|bc|bcbb|abb|cbca|bbb`

This very simple example already shows that this version of the algorithm gives longer parsed words with respect to LZ78: for example the sequence `aaaaaaaaaaaaaaa` (15 `a`'s) is parsed to `a|aa|aaa|aaaa|aaaaa` with LZ78 and `a|aaaaaaaaaaaaaa` with LZ77. This obviously has a computational cost: in this case the dominant term of the compression rate is indeed $c_n \log n$, since the pointer to the prefix can refer back to any point in the previously parsed string, and in general $c_n \log n \geq c_n \log c_n$. Anyway, the following Theorem by Ziv [72] ensures that the described algorithm is universal.

Theorem 1.2.4. *If X is an ergodic process,*

$$\frac{c_n \log n}{n} \xrightarrow[n\to\infty]{} h(X) \text{ almost certainly.}$$

We will not give the orginal proof of this result here, since it will follow from the more general theorem that will be given in the next paragrah and proved in an original way in the following.

1.2.3 Ziv-Merhav Theorem for relative entropy

Now suppose you have *two* (stationary, ergodic) information sources, X and Y. There are some obvious ways of generalizing LZ77 parsing inside a single string to *cross-parsing*, i.e., parsing a realization from X with words "coming from" a realization of Y, which can be used as the dictionary for the parsing. Let x and y be two sequences in $\mathcal{A}^*$, generated respectively by X and Y. To obtain a LZ parsing of x with respect to y, we will first identify the longest prefix of x that is also a substring of y, i.e. $\inf\{m \geq 0 \mid \exists\ i \geq 0 \text{ s.t. } x_1^m = y_i^{i+m-1}\}$. If $m = 0$, the first parsed word will simply be the first character of x. The algorithm then proceeds in the same way but starting from x_{m+1}, until all of x is parsed.

Now the question is: is there a way to use this modified version of LZ to give an approximation of the *relative entropy* between X and Y, in the same way as standard, universal LZ compressors approximate the entropy of a single source? The answer comes (at least for Markov sources) from Ziv-Merhav's Theorem [73]: the number of words of the "cross"-LZ parsing described above, appropriately scaled with the length n of the strings into consideration, goes at the limit for $n \to \infty$ to $h + d$, where h is the entropy of the process X and d is the relative entropy between X and Y.

More rigorously,

Theorem 1.2.5. *(Ziv, Merhav) If X is stationary and ergodic with positive entropy and Y is a Markov chain, with $P_n \ll Q_n$ asymptotically, then*

$$\lim_{n\to\infty} \frac{c_n(x|y) \log n}{n} = h(P) + d(P\|Q) \quad (P \times Q) - a.s.,$$

where $c_n(x|y)$ is the number of parsed words of x_1^n with respect to the sequence/database y_1^{2n}.

The original proof of this result relies on heavy probabilistic instruments; in the following section we will present a new and hopefully clearer version of the proof, that underlines the fundamental role of cross recurrences in the estimation of relative entropy.

1.3 Relative entropy via return times and match lengths

1.3.1 Definitions and asymptotical results

Let us consider again $x, y \in \mathcal{A}^{\mathbb{N}}$, two infinite strings produced respectively by X and Y.

Definition 1.9. The *cross match length* of x with respect to y is defined as

$$L_n(x|y) := \inf\{k \in \mathbb{N} \mid x_1^k \neq y_j^{j+k-1} \; \forall \; j = 1, 2, \ldots, n\}, \tag{1.6}$$

i.e. the length of the shortest prefix of x_1^∞ which cannot be found starting in the window y_1^n (the shortest non-appearing word). Note that if no match is found $L_n(x|y) = 1$.

In literature, it is frequent to find a slightly different but perfectly equivalent definition, according to which the cross match length measures the longest prefix of x that *appears* as a subsequence in y. The difference between the two versions is of exactly 1 for each match, and cannot therefore affect asymptotic results.

The cross match length is a dual quantity to the *waiting time*:

$$W_m(x|y) := \inf\{j \in \mathbb{N} \mid x_1^m = y_j^{j+m-1}\}; \tag{1.7}$$

we have indeed [69]:

$$L_n(x|y) \leq m \quad \Leftrightarrow \quad W_m(x|y) > n.$$

The asymptotic properties of cross recurrences (or equivalently of cross match lengths) between sources X and Y are governed by the entropy of the source and the relative entropy between the two sources, as stated in the following theorem [33]:

Theorem 1.3.1. *If X is a stationary ergodic source with $h(X) > 0$ and Y is a Markov chain, with $P_n \ll Q_n$ eventually,*

$$\lim_{m\to\infty} \frac{\log W_m(x|y)}{m} = h(P) + d(P\|Q), \quad (P \times Q) - a.s..$$

Equivalently, in terms of match lengths:

$$\lim_{n\to\infty} \frac{L_n(x|y)}{\log n} = \frac{1}{h(P) + d(P\|Q)}, \quad (P \times Q) - a.s..$$

The proof of Theorem 1.3.1 is given in [33, 34] and consists of two steps:

1. The first step is a generalization of Shannon-McMillan-Breiman's Theorem due to A.R. Barron [1], which ensures that

$$\lim_{n\to\infty} \frac{1}{n} \log \frac{1}{Q(x_1^n)} = h(P) + d(P\|Q), \qquad P\text{-a.s.} \tag{1.8}$$

Here we will give a simple proof of (1.8) in the case when both X and Y are Markov chains and $P_n \ll Q_n$ eventually (a generalization to Markov processes with memory k is straightforward; see [1] for the general case). Let us denote with p_{ab} and q_{ab}, $a, b \in \mathcal{A}$, the transition probabilities for P and Q respectively. Given a sequence x_1^n from P, let n_{ab} be the number of occurrences of the couple ab in x_1^n; then

$$\log Q(x_1^n) = \log Q(x_1) + \sum_{ab\in\mathcal{A}^2} n_{ab} \log q_{ab}$$

Since by Theorem 1.1.1 $n_{ab}/n \to P(ab) = P(a)p_{ab}$ P-a.s, we simply have that

$$\lim_{n\to\infty} \frac{1}{n} \log \frac{1}{Q(x_1^n)} = -\sum_{ab} P(a)p_{ab} \log q_{ab} \quad P\text{-a.s..}$$

On the right hand side we recognize

$$\begin{aligned} -\sum_{ab} P(a)p_{ab}\log q_{ab} &= -\sum_{ab} P(a)p_{ab}\log\left[\frac{q_{ab}}{p_{ab}}\right] + \\ &\quad -\sum_{ab} P(a)p_{ab}\log p_{ab} \\ &= h(P) + d(P\|Q). \end{aligned}$$

This concludes the proof of (1.8) for Markov chains.

2. The second step relates almost sure asymptotic properties of recurrence times to the measure of the recurrent sequence. More precisely:

$$\log W_n(x|y) - \log\frac{1}{Q(x_1^n)} \xrightarrow[n\to\infty]{} 0 \qquad Q\text{-a.s. in } y \text{ and } \forall\, x. \tag{1.9}$$

We stress here that the notion of waiting time (or *hitting time*) has recently attracted lot of attention in the dynamical systems community. Some of the idea developed in that area of research can be fruitfully transposed in the present context. Indeed consider a probability space $(\Omega, \mathcal{S}, P)$ and a partition $\mathcal{P} = \{P_a\}_{a\in A}$ of Ω. The random variable $X_{\mathcal{P}}(x)$ defined as $X_{\mathcal{P}}(x) = a$ if $x \in P_a$, together with a measure-preserving transformation T, defines a stochastic process, the so-called $(T, \mathcal{P})$-process (for precise definitions see for example [60]). Poincaré recurrence theorem states that almost every point of $B \subseteq X$, with $P(B) > 0$, will eventually return in B. Kaç lemma quantifies this return by relating the average return time to the measure $\mathbb{P}(B)$ of the reference set. This result was later improved and extended in various directions. It is worth to mention the following result (see for example [57] and references therein), particularly relevant for our purposes: take a family of nested sets B_n with $\mathbb{P}(B_n) \to 0$ (precise conditions on B_n can be found in [57]); then for almost every x in B_n the first return time $\tau_{B_n}(x)$ in B_n has the property: $\log \tau_{B_n}(x) - \log\frac{1}{P(B_n)} \xrightarrow[n\to\infty]{} 0$.

This result was recently extended (see for example [21] and reference therein) to deal with the waiting time, that is, the number of iterations

of T that a point $x \in \Omega$ needs to enter a given set B for the first time (note that the waiting time is equivalent to the return time if x is forced to start in the reference set B). In this setting the search for a match of the type (1.9) can be rephrased as follow: first consider the $(T, \mathcal{P})$-process equivalent to the stochastic process under study. Let B_n be the family of cylinders induced by the sequence y: $B_n := [y_1, y_2, \cdots, y_n]$. Note that the condition $P(B_n) > 0$ is guaranteed by the assumption that $P_n \ll Q_n$ eventually. The quantity $W_n(x|y)$ defined in (1.7) is then equivalent to the waiting time for a first passage in B_n and thus (1.9) follows from the more general result mentioned above. We stress here that dealing with the cylinders B_n is much simpler than the general case (for example, balls) treated in the above mentioned references; we expect that these stronger results derived in dynamical systems could be fruitfully applied in the setting of information theory, providing effective tools for the derivation of novel techniques.

1.3.2 Relative entropy via return times and match lengths

In order to provide an useful tool that can stand as a starting point for a meaningful definition of a (pseudo)-distance between two finite sequences, we need to somehow modify the results in Theorem 1.3.1, that deal with *infinite* sequences. In order to cope with the *finite* length sequences x_1^n and y_1^{2n}, we define a *truncated* version of the match length defined above:

$$\tilde{L}_n(x|y) := \min\{L_n(x|y), n\}. \tag{1.10}$$

Now, let $\sigma : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ be the left shift and define recursively the sequence of indices $\{l_1, l_2, \ldots, l_c\}$:

$$
\begin{aligned}
l_1 &:= \tilde{L}_n(x|y) \\
l_2 &:= \tilde{L}_{n-l_1}(\sigma^{l_1}(x)|y) \\
l_3 &:= \tilde{L}_{n-(l_1+l_2)}(\sigma^{l_1+l_2}(x)|y) \\
&\dots \\
l_c &:= \tilde{L}_{n-(l_1+l_2+\dots+l_{c-1})}(\sigma^{l_1+\dots+l_{c-1}}(x)|y), \qquad (1.11)
\end{aligned}
$$

where $\sigma^{l_i}(x) = x_{l_i}^n$, with $1 \leq i \leq c$.

The LZ parsing of the sequence x_1^n with respect to y_1^{2n} that we described in the previous paragraph can now be represented as follows:

$$x_1^n = \{x_1^{l_1}, x_{l_1+1}^{l_1+l_2}, \dots, x_{n-(\sum_{i=1}^{c-1} l_i)}^n\}, \qquad (1.12)$$

Note that this parsing is the one that Ziv and Merhav define in [73] where the length of the parsed word is exactly the match length[1]. Here, as discussed above, $c_n = c_n(x|y)$ is the number of parsed words of x_1^n with respect to the sequence/database y_1^{2n}, and it plays a fundamental role in the relationship with the relative entropy, as stated by Theorem 1.2.5.

We are now ready to prove the Theorem of Ziv-Merhav.

Proof. Let us suppose that $h(P)$ and $d(P\|Q)$ are not both equal to zero (otherwise $c_n = 1 \; \forall \; n$ and the theorem becomes obvious). First note that

$$\frac{n}{c_n \log n} = \frac{1}{c_n} \sum_{i=1}^{c_n} \frac{l_i}{\log n},$$

where every l_i is of the form $\tilde{L}_k(\sigma^j(x|y))$ for some $j, k \in \mathbb{N}$ (see definition in (1.11)). Observe that, for every x, y, there is a $N \in \mathbb{N}$ such that $\tilde{L}_k(\sigma^j(x)|y) = L_k(\sigma^j(x)|y)$ for all $k \geq N$; indeed, $\frac{\tilde{L}_n(x|y)}{\log n} \leq \frac{L_n(x|y)}{\log n}$ which goes to $1/(h+d)$ for $n \to \infty$ by Theorem 1.3.1. Since, moreover, Theorem 1.3.1 holds for a.a. x and y, $L_k(\sigma^j(x)|y)$ has the same asymptotic behavior as $L_k(x|y)$ for all $j \in \mathbb{N}$

[1] except maybe the last word, but we neglect this detail, since we are interested in asymptotic results.

and for a.a. x, y. We can therefore substitute each $\frac{l_i}{\log n}$ with $\frac{L_n}{\log n} + \varepsilon_i(n)$, with the corrective term $\varepsilon_i(n) \xrightarrow[n\to\infty]{} 0$, so that:

$$\frac{L_n}{\log n} + \min_{i=1,\dots,c} \varepsilon_i(n) \leq \frac{n}{c_n \log n} = \frac{1}{c} \sum_{i=1}^{c} \frac{L_n}{\log n} + \varepsilon_i(n) \leq \frac{L_n}{\log n} + \max_{i=1,\dots,c} \varepsilon_i(n)$$

and thus, taking the limit for $n \to \infty$,

$$\frac{n}{c_n \log n} \xrightarrow[n\to\infty]{} \frac{1}{h(P) + d(P\|Q)}.$$

□

In the case when X and Y are the same source ($P = Q$), Theorem 1.2.5 gives another proof of the universality of the LZ77 compression algorithm described in section 1.2.1, and precisely of the version with fixed database, defined in [67]. Our proof is different from the the one proposed there by Wyner and Ziv and later by Ornstein and Weiss in [51].

Another proof of the Theorem of Ziv-Merhav, that makes use of the Burrows-Wheeler transform [10], can be found in [11].

Chapter 2

Applications: authorship attribution

Ordinarily when an unsigned poem sweeps across the continent like a tidal wave whose roar and boom and thunder are made up of admiration, delight, and applause, a dozen obscure people rise up and claim the authorship.
- M. Twain, *Is Shakespeare dead?* -

2.1 *Stylometry*, from words to n-grams

Why and how should authorship attribution (or *stylometry*, as it is sometimes called) be the object of a mathematical study? The idea of applying quantitative (not always mathematically founded) ideas to the problem of recognizing the author of an anonymous or apocryphal text is not new, but dates back at least to the end of the 19^{th} century, when two studies by the mathematician A. De Morgan [19] and the geophysicist T. C. Mendenhall [48] proposed to calculate the average lengths of words in the works of different writers and to compare them in order to establish authorship.

During more than one century of history, a large variety of methods with different origins were applied to authorship attribution problems, from scientists of many different fields; for an extensive review see for example [25] and the more recent [37, in Italian], [23] and [62]. Disregarding those methods

that are too distant from our approach, a few choices are needed to define an authorship attribution method.

First of all, we need to choose the quantity that we want to compute in the texts or, as it is usually called, the *feature* to extract. There is a huge variety of possibilities, with different origins and approaches to the written text. Following the scheme in [62], to which we refer for an exhaustive list of references, a possible classification of features is the following:

1. *character-level features*: character frequencies, character $n-$grams...

2. *lexical features*: word frequencies, vocabulary richness, errors...

3. *syntactic features*: sentence and phrase structure, part-of-speech distribution...

4. *semantic features*: synonyms, ...

Note that as the order in the list proceeds, the proposed methods use more and more of the structure of the text, correspondingly increasing the complexity of the techniques and of the instruments they require.

Even if lexical or syntactic methods are still frequent in literature, several works of the last decade adopted instead the simpler approach of carachter-level features. In this case the text is considered merely as a sequence of symbols, as it is quite natural for non-linguists. Indeed, this point of view was adopted both by Markov [42, 43] and by Shannon [59] in their original works: in both cases, the words as basic components of the text have no more meaning than other aggregates of symbols, while the statistics of sequences of n consecutive characters (the so called *n-grams*) appear naturally as the fundamental object of investigation. From this point of view, the link between the study of written texts and the results on symbolic sequences in chapter 1 becomes clearer: the analysis of texts is deprived of historical and phylological details, syntactic or lexical considerations are in general avoided, and we are therefore reduced to the automatic extraction of information to be used

for stylistic or semantic classification. Here is where statistical measures as the entropy and divergence come into play, as we will see in the following.

The "n-gram approach" has attracted more and more interest in the last decade in the field of authorship attribution: R. Clement and D. Sharp, for example, proposed in 2003 [14] a method based on n-gram frequencies, whereas in 2001 D. V. Khmelev and F. J. Tweedie [32] published some results obtained by considering texts as first order Markov chains, i.e. by calculating a (empirical) one-step transition matrix from the reference texts of an author and then using it to establish the probability for a given anonymous text to have been written by that author. A validation of non-lexical methods came from the results of the *Ad-hoc Authorship Attribution Competition* (AAAC), an attribution contest launched by P. Juola in 2003 [29]: one of the best global results for the 13 data sets of the competition was obtained by V. Kešelj using a metric method, once again based on n-gram frequencies, which was the starting point of part of our research (see paragraph 2.2.2). More recently also E. Stamatatos [61] used $n-$grams to classify a corpus of Greek newspaper articles, and J. Grieve [23] found them to be the most effective feature in a large-scale comparison of a number of lexical and syntactic authorship attribution methods.

Again, the key idea, common to all of these approaches, is to consider a text "just" as a symbolic sequence, not taking into consideration either the content of the text or its grammatical aspects: letters of the alphabet, punctuation marks, blank spaces between words are just abstract symbols, without a hierarchy. Using the language of chapter 1: a text is seen as an element of $\mathcal{A}^* = \cup_{n\in\mathbb{N}}\mathcal{A}^n$, the set of finite sequences over the finite alphabet $\mathcal{A}$. The alphabet can be any finite set, and it is in general quite large; for the Gramscian corpus which will be described later, for example, $\mathcal{A}$ is made up of 84 symbols: the 21 letters of the Italian alphabet (upper and lowercase, with and without accent), together with some letters of foreign alphabets; the digits from `0` to `9`; the commonest punctuation marks and the blank space.

Following the scheme of information theory as developed in the previous chapter, symbolic sequences can be thought as being generated sequentially by an information source according to some probability distribution; such distribution is generally (and in particular in our case) unknown, but texts can be considered as "randomly generated samples" of the source. For the case of authorship attribution, we would like to identify the *author* as the information source and be able to reconstruct the distribution of the source/author by measuring some quantities in the texts he wrote; or, to say it in a more pictoresque way: we would like to give a quantitative characterization of an author's style.

Once that we have chosen "what to count", we are faced with a second important choice: how to count it, i.e., how to use the information that we have extracted from the texts to establish the authorship. And again, there is a vast plethora of methods, many of which use advanced techniques like machine learning.

The approach followed in our experiments consists in synthesizing as a single quantity the difference/dissimilarity observed by measuring the chosen features in the texts. This value will be a measure of the proximity of two texts or of a text and an author; in other words, we would like to use these measures to define on the set of symbolic sequences a *distance* that can account for the stylistic similarities between authors.

2.2 The Gramsci Project

2.2.1 Description of the project and corpora

Starting in 2006, a group made up of researchers from the Mathematics Department of the Universities of Bologna (M. Degli Esposti, C. Basile) and Rome - *La Sapienza* (D. Benedetto, E. Caglioti), together with the linguist M. Lana from the University of Western Piedmont in Vercelli, started

working on a project proposed by the *Fondazione Istituto Gramsci*[1], based in Rome. The idea was to recognize the authorship of a large number of short newspaper articles written by Antonio Gramsci[2] and his coworkers during the first decades of the XX century. The *Istituto Gramsci* is publishing a complete edition of the works of the author (*Edizione nazionale degli scritti di Antonio Gramsci*[3]), trying to include also a number of those articles that he left unsigned, a practice which was not uncommon at that times.

The *Gramscian corpus*, as we will call it in the following, is very interesting from the point of view of a stylometric study. Indeed, in order to be sure that the recognition is based on authorship and not on other factors, a good corpus has to be homogeneous in as many respects as possible (cf. the discussion in [23]); this is exactly the case of our corpus, that is homogeneous from the point of view of the publishing time (a couple of decades), the genre (articles published on a limited number of newspapers), the audience (the readers of Communist newspapers) and, more generally, the cultural background shared by both the authors and the readers of those texts. Even more important, it is interesting that the articles deal with common subjects: when experimenting a quantitative authorship attribution method, indeed, it is crucial to distinguish its results from a possible distinction by subject. In the case of the Gramscian corpus, the texts deal mainly with contemporary political events.

Last, but not less important, our texts are *short*, ranging from one to around fifteen thousand characters: discriminating between short texts is much tougher than dealing with whole novels, due to the lack of statistics; on the other side, most attribution cases from the real world require the capability of dealing with short documents, and our research goes precisely in that direction.

[1] `http://www.fondazionegramsci.org/`

[2] Antonio Gramsci (Ales, 1891 - Rome, 1937) was a famous Italian politician, philosopher and journalist, and one of the founders of the Communist Party of Italy.

[3] `http://www.fondazionegramsci.org/ag_edizione_nazionale.htm`

2.2.2 The n-gram distance

The first method we used is probably one of the simplest possible measures on a text, and it has a relatively short history in published bibliography.

After a first experiment based on bigram frequencies presented in 1976 by W. R. Bennett [9], V. Kešelj et al. published in 2003 a paper [30] in which n-gram frequencies were used to define a similarity distance between texts. First, they define a so-called *profile* for each author A, built in this way: once the value of n has been fixed, usually between 4 and 8, n-gram frequencies are calculated using all the available texts by author A. Then these n-grams are disposed in decreasing order by frequency and only the first L are taken into consideration, where L is a further parameter to set. The same operation (n-gram frequency extraction and ordering) is then repeated on the unknown text x for which attribution is sought.

We call ω an arbitrary n-gram, $f_x(\omega)$ the relative frequency with which ω appears in the text x, and $f_A(\omega)$ the relative frequency with which ω appears in author A's texts, $D_n(x)$ the *n-gram dictionary* of x, that is, the set of all n-grams which have non-zero frequency in x, and $D_n(A)$ the n-gram dictionary of all author A's texts. With these notations a text x can be compared with a profile A through the following formula, which defines a measure of the proximity between text x and author A:

$$d_n^K(x, A) := \sum_{\omega \in D_n(A) \cup D_n(x)} \frac{(f_A(\omega) - f_x(\omega))^2}{(f_A(\omega) + f_x(\omega))^2}. \tag{2.1}$$

In presence of the L parameter, the sum is restricted to the first L $n-$grams in order of decreasing frequency. The text x is thus attributed to the author A for which the distance is minimal. The authors of [30] assert that the inspiration for this formula came from the paper [9] by Bennet, who used as a (dis)similarity indicator the distance defined simply as the sum of the squares of the differences between frequencies in A and x, i.e. the squared Euclidean distance between frequency vectors.

Note that in formula (2.1), in contrast with what happens for the Euclidean distance, each term of the sum is weighted with the inverse of the

square of the sum of the frequencies of that particular n-gram in A and in x, so that terms related to "rare words", i.e., n-grams with lower frequencies, give a larger contribution to the sum. In this way, for example, a difference of 0.01 for an n-gram with frequencies 0.09 and 0.08 in the two profiles will have a lower weight than the same difference for an n-gram with frequencies 0.02 and 0.01. It is also useful to underline that $d_n^K(x, A)$ is indeed a *pseudo-distance*, for instance it does not satisfy the triangular inequality. In the following, however, we will conform to the accepted practice of calling all such functions "distances", with a small and unimportant lack of mathematical rigor.

Kešelj and his coworkers tested the effectiveness of their method on different text corpora: literary works by 8 English authors from different ages; newspaper articles of 10 different authors, written in modern Greek; some novels on martial arts by 8 modern Chinese writers. As reported in the cited work [30], the final results are quite satisfactory and they reach or surpass in almost every case (with the only exception of the Chinese corpus) the ones obtained with the methods experimented before by Kešelj and other researchers on the same text sets. It is worth observing, though, that the dependance on one or two parameters (n and possibly L) puts forward a methodological problem: how to choose n and L for a *real* attribution problem, in which the solution is not known? In the very brief paper [31] Kešelj and Cercone suggested indeed a suitable *weighted voting* to answer this question: this is the so-called *Common N-Grams Method with Weighted Voting* they used in the AAAC, see [29].

For our experiments on Gramsci's articles recognition we partially used Kešelj's ideas, adjusting them to fit our particular scenario, which has peculiar characteristics when compared to other attribution problems (e.g. the ones of AAAC). A first aspect is that our aim is "just" to determine whether a text was written by Gramsci or not, and not to establish the attribution of the specific author of every text; this feature is an element of great simplification if compared to a generic attribution problem with many possible

authors.

In a preliminary tuning stage we used 100 texts, 50 by Gramsci and 50 by the 17 other authors listed in Table 2.1 together with some data concerning the length of the available articles for each author. Some modifications of Kešelj's method followed directly from the examination of the data set. First of all, we decided not to merge to a single profile all the texts of a reference author but to calculate the distance between every single pair of available texts. Indeed, in this case building the authors' profiles as in Kešelj's method would be in contrast with the characteristics of the 100 articles by Gramsci and coworkers, where the subdivision of the texts among the different authors is strongly heterogeneous (see Table 2.1), so that merging all texts of an author in a single profile we would have obtained profiles with very different statistical meaning: note, for example, the large disparity between the total lengths of available texts by Gramsci and by Viglongo.

Furthermore, because of the shortness of the articles and of the choice of comparing them individually, the L parameter became unuseful and it was necessary to consider all the possible n-grams with nonzero frequency. In a single text and for large n, indeed, as can be seen from the example in Table 2.2, most of the n-grams appear just once, so that considering just the L more frequent ones would be the same as arbitrarily choosing L n-grams.

Ultimately, in order to eliminate the strong dependance of Kešelj's formula on the length of the texts into consideration, the distance is divided by the sum of the number of n-grams in the two texts; the resulting formula is the following, for two texts $x, y \in \mathcal{A}^*$:

$$d_n(x,y) = \frac{1}{|D_n(x)| + |D_n(y)|} \sum_{\omega \in D_n(x) \cup D_n(y)} \left(\frac{f_x(\omega) - f_y(\omega)}{f_x(\omega) + f_y(\omega)} \right)^2 . \qquad (2.2)$$

From now on we will call *n-gram distance* the one defined in (2.2), unless otherwise stated. Again, d_n is a pseudo-distance, since it does not satisfy the triangular inequality and it is not even positive definite: two texts x, y can be at distance $d_n(x,y) = 0$ without being the same.

author	number of articles	total length of the articles	mean length of the articles
Antonio Gramsci	50	326843	6536.9
Palmiro Togliatti	11	91334	8303.1
Amedeo Bordiga	7	47894	6842.0
Angelo Tasca	5	48684	9736.8
Leo Galetto	4	18623	4655.7
Adolfo Giusti	4	14346	3586.5
Giuseppe Bianchi	3	12928	4309.3
Attilio Carena	3	23556	7852.0
Giacinto Menotti Serrati	3	12852	4284.0
Alfonso Leonetti	2	16514	8257.0
Gino Castagno	1	8146	8146.0
C. D.	1	5612	5612.0
Alessandro De Giovanni	1	6700	6700.0
C. F.	1	2659	2659.0
Ottavio Pastore	1	4176	4176.0
Mario Santarosa	1	5053	5053.0
Umberto Terracini	1	9432	9432.0
Andrea Viglongo	1	7450	7450.0

Table 2.1: Total and average character length of the articles used in the preliminary phase, by author.

The results of the attribution of the 100 texts, obtained by assigning to each unknown text the author of its nearest neighbour according to the distance d_n, are plotted in figure 2.1. The length of n-grams varies along the horizontal axis, with n from 1 to 10; two symbols correspond to each value of n: the circle marks the number of Gramscian texts which are correctly attributed to him by the method (true positives), while the triangle indicates the number of non-Gramscian texts which are correctly recognized as such (true negatives).

For the following experiments we chose $n = 8$: for this value, indeed,

n	total n-grams	n-grams appearing only once	percentage of n-grams appearing only once
1	62	9	15%
2	416	107	26%
3	1576	689	44%
4	2948	1805	61%
5	3960	2948	74%
6	4611	3806	83%
7	5030	4405	88%
8	5297	4806	91%
9	5480	5086	93%
10	5611	5294	94%
11	5707	5453	96%
12	5777	5565	96%
13	5837	5660	97%
14	5888	5741	98%
15	5931	5807	98%

Table 2.2: Number and percentage of occurrences of n-grams appearing only once in the text *g_27*.

we achieved the best attribution results (41 texts out of 50) without loosing too much in precision (only 5 false positives). We will comment later on the implications of such a choice for n.

These first results were obtained by taking into consideration only the first neighbour of each text. Such a choice ignores the fact that the reference set contains as many as 100 different articles with which one can compare the given "unknown" text. This suggests some questions:

- what can we expect about the distance of an article by Gramsci from the 49 other texts by him?
- will these 49 articles be "nearer on the average" to the text in consideration?

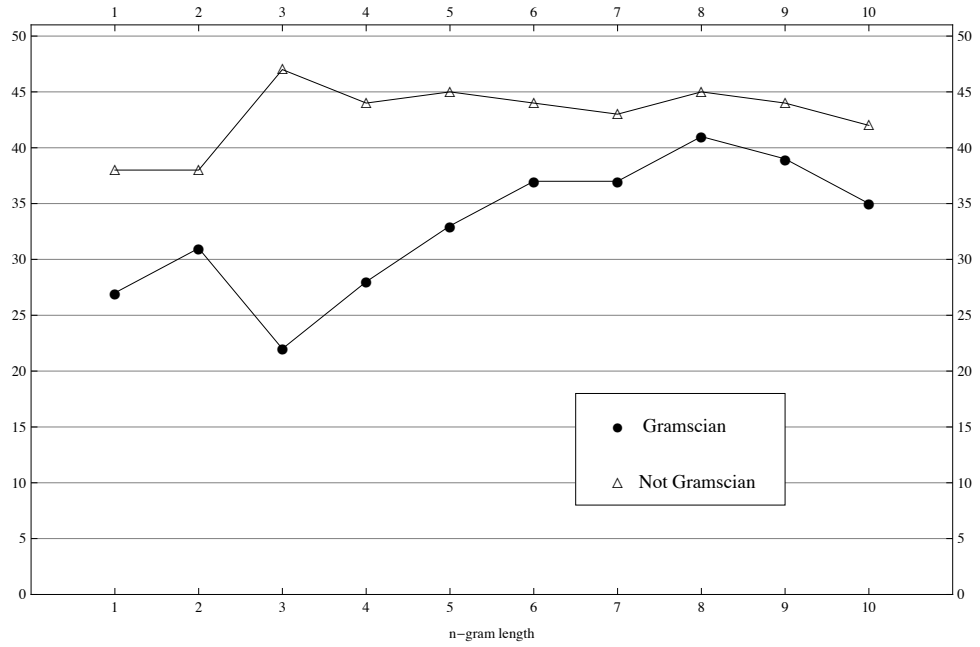


Figure 2.1: Number of correctly attributed Gramscian and non-Gramscian texts with the first neighbour, over the 100 texts of the training corpus.

- is it possible to consider conveniently also the distances from all the other reference texts, not only the first neighbour?

Some of these issues, especially the first, will be further discussed later, in paragraph 2.2.6. Trying to give a first answer to such questions, we defined for a text x a *Gramscianity index* $g(x)$ in the following way: all the reference texts are listed in order of growing distance from the text x; the j-th text by Gramsci in the list is given the score $k(j)/j$, where $k(j)$ is its rank in the list; the Gramscianity index $g(x)$ is the sum of the scores of the 49 texts by Gramsci which appear in the list. The *non-Gramscianity index* $ng(x)$ of text x is defined similarly as the sum of the corresponding scores for the first 49 texts not by Gramsci.

The Gramscianity index will be lower as long as the unknown text is nearer to the group of Gramscian texts ($ng(x)$ has the same property for non-Gramscian texts). The text x is therefore attributed to Gramsci if its Gramscianity index $g(x)$ is lower than its non-Gramscianity index $ng(x)$.

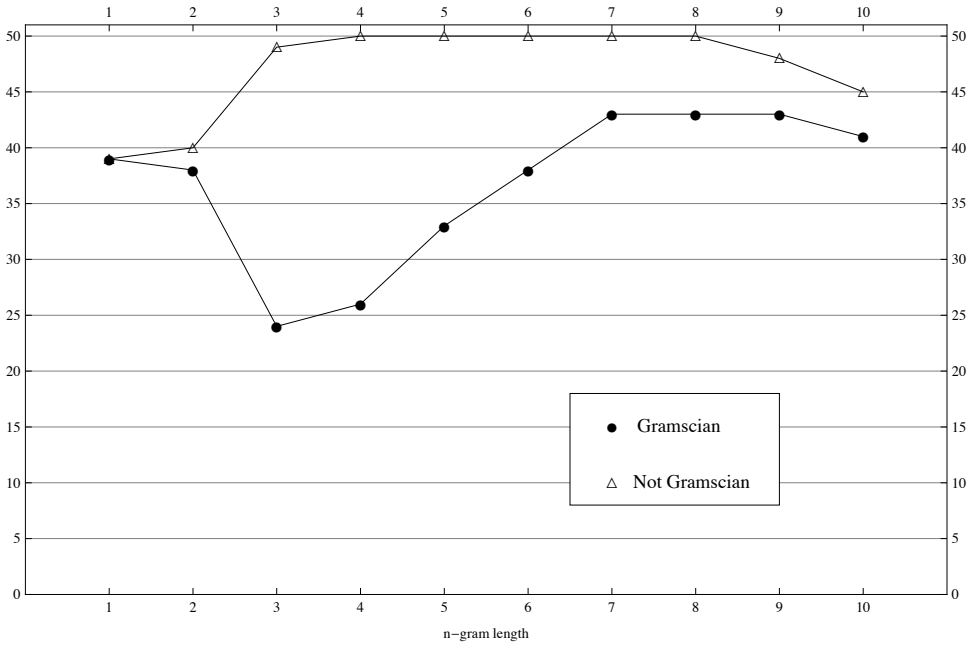


Figure 2.2: Attributions using Gramscianity and non-Gramscianity index, for the 100 texts of the training corpus.

Figure 2.2 illustrates, with the same conventions used for figure 2.1, the results obtained for the 100 text corpus using the index method with n-gram length from 1 to 10. Even in this case, the results suggest $n = 7$ or $n = 8$ as the best choices for the parameter; for such values, indeed, we have the best results for the recognition of texts by Gramsci (43/50 texts) and no false positive.

The use of these indices has also another advantage: their difference gives a natural measure of the reliability of the attribution. More precisely, given an article x to attribute, if $g(x)$ and $ng(x)$ are the Gramscianity and non-Gramscianity indices defined above, the number

$$v(x) = \frac{ng(x) - g(x)}{ng(x) + g(x)} \tag{2.3}$$

lies always between -1 and 1: a value near to 1 (or -1) gives a strong attribution to Gramsci (or to "non-Gramsci"), while values near to 0 are a mark of great undecidability.

The value of the vote v for each of the 100 texts of the corpus is illustrated in figure 2.3: it is easy to notice that some texts, for example g_{03}, g_{08} and n_{39}, have a much weaker attribution than g_{04} or n_{01}.

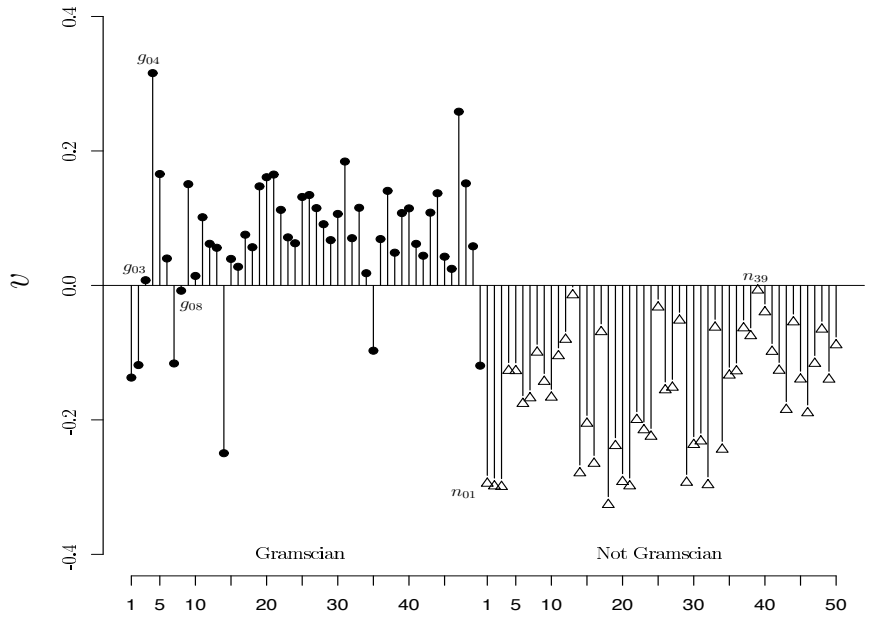


Figure 2.3: Attribution of the 100 texts with measure of attribution reliability, using the Gramscianity and non-Gramscianity indices defined in the text.

2.2.3 Entropy and compression: the *BCL* method

Shannon's information theory, as described in chapter 1, has a rigorous and consistent formulation only for well defined mathematical objects: we have seen how the stationarity and ergodicity of the source play a fundamental role in its development. Anyway, it is quite natural to use it also in the field of text analysis. Shannon himself, indeed, estimated with an experiment that the average quantity of information of the source "English language" is between 0.6 and 1.3 bits per character. Though the entropic characteristics

of an author's writing are certainly interesting, an approximated value of entropy *per se* is not very useful for the attribution problem, as can be seen in Table 2.3, where the compression rate obtained with an LZ compressor is listed for various authors of Italian literature: it would not be possible to distinguish Dante's works from Boccaccio's only based on this measure and, on the other hand, different works by the same author can have very different values of entropy.

author	work	compression rate
Dante	Commedia	3.2
	De Vulgari Eloquentia	3.0
	Convivio	2.7
Boccaccio	Decamerone	2.8
Petrarca	Canzoniere	3.1

Table 2.3: Compression rates in bits per character of some texts from Italian literature.

Moving now from a single source (author) to the comparison between two sources, relative entropy can be considered as a very powerful tool to quantify their difference: it is indeed reasonable to expect that the relative entropy of two texts by Boccaccio is smaller than the one between a text by Boccaccio and a text by Petrarca. Moreover, relative entropy can be computed effectively using compression algorithms, as we have seen in 1.

As we have seen, some compression algorithms, and the ones in the LZ family in particular, allow indeed to obtain an estimate of the relative entropy between two texts, and hence to measure their closeness. Various methods based on Ziv-Merhav's theorem and similar ideas have been proposed and used on specific problems in the fields of biological sequence analysis and of authorship attribution; here we cite, with no pretension of completeness, the works of M. Li et al. [39], P. Juola [28], W.J. Teahan [65], O.V. Kukushkina, A.A. Polikarpov and D.V. Khmelev [35], and D. Benedetto, E. Caglioti and V. Loreto [8].

In [8] it has been proposed to estimate the relative entropy as follows (see also [35],[41]). Suppose you compress the text $y + x$, that is, the text obtained by attaching text x to text y. The compression algorithm, being sequential, will code first all the characters of y and then will start coding those of x, looking for the strings in the part which it has already read, i.e. in the text y. The more similar the two texts are, the longer will be the strings in x which are found in y, and therefore the more effective will be the compression of the whole file. The compressor, in fact, in this case will take advantage not only of the redundancy within the single texts, but also of the one between the two texts, improving the compression rate. The difference between the lengths of the compressed versions of $y + x$ and of y, divided by the length of x, is a measure of the relative entropy of text x with respect to y (for a detailed analysis of the compression of attached files see [56]).

It is actually possible to implement the method described above using *winzip* or *gzip*, and the results obtained are reasonable. However, in LZ77-based compressors, the coding phase is followed by another one in which suitable algorithms re-code the couples of numbers in order to optimize compression. Benedetto, Caglioti and Loreto have therefore developed a program, called *BCL*, in which this re-coding is optimized, in a way similar to *gzip*, to improve attribution skills, and where the repeated strings are searched only in the first file, in the attempt to apply Ziv-Merhav's theorem. Note that this is exactly the cross compression algorithm we discussed in 1.2.3.

Here we used this method, adapting it to the Gramscian problem. The results for the 100 texts by Gramsci and coworkers used in the preliminary tuning stage were not good enough: 32 true positives and 14 false positives. The point is that the entropic method is strongly sensitive to the size of reference texts. In general, all the methods based on the comparison of single texts tend to choose the nearest neighbour of the unknown text among those of larger size. Long texts, indeed, are relatively richer in statistics and information, and are therefore likelier to have common characteristics with the test documents. On the other hand, if for the n-gram method the texts

appearing at first ranks in the attributions have a size about 1.5 times the average for the 100 texts, for the entropic method this ratio is above 2.

Therefore, we proceeded by using "reassembled" texts for the comparison: the reference corpus, for example Gramscian texts, was first merged into a single large file and then cut into many parts of equal length (loosing the original partition into articles). These new files of equal size became the new reference texts. The results were sensibly better; we show them in figure 2.4, varying the standard length of reassembled texts for the reference corpus.

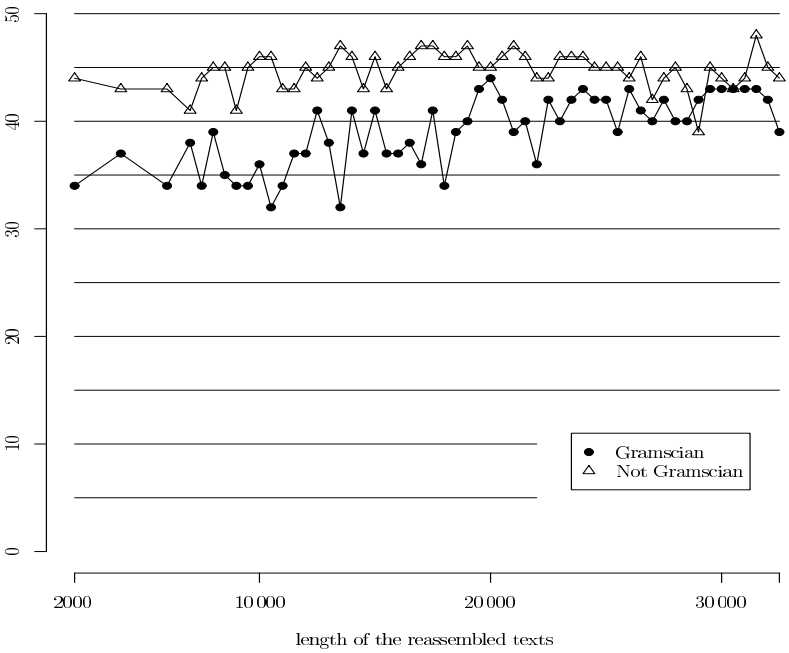


Figure 2.4: Attribution of Gramscian and non-Gramscian texts with BCL method, related to the length of reassembled texts.

We then applied also here the voting technique described for n-grams, but limiting the sum to the first three Gramscian and non-Gramscian texts in the ranking. With a length of 29 500 bytes we had the best results: 46 true positives and one false negative.

2.2.4 The overall procedure and the experiments

The overall attribution strategy was hence based on the two methods described above:

- 8-grams with vote extended to all the reference texts;
- relative entropy with reassembled texts 29 500 characters long and vote for the first three texts in the ranking.

The two methods are based upon completely different principles. On the other hand, one can fear that they give in fact the same information, adding nothing to the accuracy of the global method. We have therefore made sure, with suitable methods, of the statistical independence of the rankings of the texts ordered according to the two distances.

We finally attributed to Gramsci only the texts that both methods assigned to him. Moreover, both techniques give a numerical value for the attribution (see eq.(2.3)) so that it is possible (and very useful) to give a graphical representation of the overall results, as the one shown in figure 2.5.

The *Gramscianity index* obtained with the n-gram method is plotted on the horizontal axis: positive values correspond to the attribution to Gramsci, negative values to "non-Gramsci". The rightmost points are the texts for which the attribution to Gramsci is more certain, the leftmost are those for which the method suggests with greatest certainty an attribution to authors different from Gramsci. On the vertical axis we show the value of the analogous index given by the relative entropy method; in this case advancing from bottom up means moving from suggested non-Gramscian texts to suggested Gramscian ones.

The absence of triangles in the first quadrant means that there are no false positives (no wrong attributions to Gramsci). The number of texts correctly attributed to Gramsci is 43, the 86% of the total. In the second quadrant lie the texts attributed to Gramsci by the relative entropy method but not by the n-gram method: among them the Gramscian texts g_{07}, g_{08} and g_{35}.

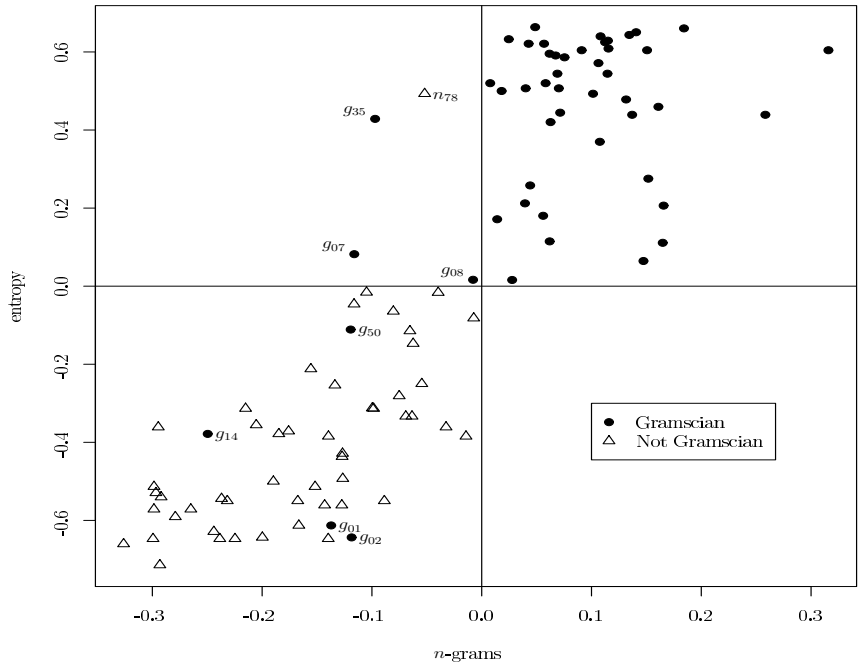


Figure 2.5: The attributions for the 100 texts for the training, at the end of the preliminary stage.

There is no text in the fourth quadrant: these would be those attributed to Gramsci by the n-gram method but not by the entropic method. Finally, the third quadrant contains the texts not attributed to Gramsci by either method. Among them the Gramscian texts g_{01}, g_{02}, g_{14} and g_{50}.

Next, we applied this procedure to the attribution of 40 additional articles we received by the scientific committee of the National Edition during a *blind* test. The application of the method to these 40 new articles gave the results shown in figure 2.6, obtained with the same procedure used in figure 2.5 for the 100 tuning texts: the abscissa of each point represents the vote given by the n-gram method, on the ordinate there is the value of the vote for the relative entropy method; the texts which both methods attribute to Gramsci lie therefore in the first quadrant.

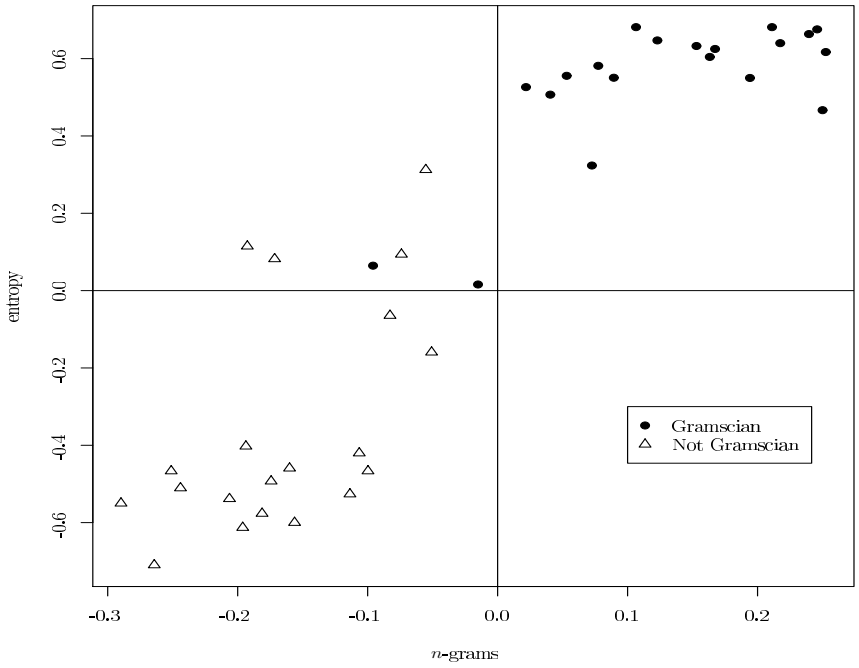


Figure 2.6: Attributions for the 40 texts of the blind test.

The results of this blind test were very good: 18 Gramscian texts out of 20 were correctly attributed to Gramsci, that is the 90%, with no false positives. There are just two Gramscian texts not recognized by the n-gram method, corresponding to the two circular marks in the second quadrant: one is a very short text, consisting of few lines, and hence objectively quite difficult to attribute, while the other one shows no singular characteristics.

2.2.5 n-gram statistics

An interesting object of investigation concerning the 8-gram distance is the statistical distribution of the 8-grams in the 100 training articles. Let us consider the big text made up of the 50 training texts by Gramsci and the one made up of the 50 training texts by non-Gramscian authors. In the following table we summarize some statistical properties of the 8-grams in

this data set.

	number of 8-grams with their multiplicity	number of distinct 8-grams	number of 8-grams with 1 occurrence
Gramscian	326 836	186 986	141 206
Not Gramscian	335 951	194 594	149 212
all	662 787	327 510	237 276

Table 2.4: 8-gram statistics for the Gramscian corpus.

Moreover, if ω is an 8-gram chosen randomly in Gramscian texts, the probability that ω does not appear in any non-Gramscian text is 0.492, and the probability that it does not appear throughout Gramscian texts, if it is chosen in non-Gramscian ones, is 0.498. Note that the term $\frac{f_x(\omega)-f_y(\omega)}{f_x(\omega)+f_y(\omega)}$ in formula (2.2) for the n-gram distance is 1 in case either $f_x(\omega) = 0$ or $f_y(\omega) = 0$. The previous considerations suggest that a relevant part of d_n is due to the contribution of 8-grams which are present only in one of the two files; namely for the 100 training texts this part is, on average, the 92% of the distance. In figure 2.7 we show the mean part of d_n which is due to n-grams with frequency 0 in one of the two files, as a function of n.

These considerations suggest to test another, simplified n-gram distance:

$$d_n^s(x,y) = \frac{|D_n(x) \triangle D_n(y)|}{|D_n(x)| + |D_n(y)|},$$

or, equivalently,

$$d_n^s(x,y) = \frac{|D_n(x) \triangle D_n(y)|}{|D_n(x) \cup D_n(y)|},$$

where $D_n(x)$ and $D_n(y)$ denote, as before, the n-gram dictionaries for texts x and y respectively and Δ is the symmetric difference: $A \triangle B = (A \setminus B) \cup (B \setminus A)$. For $n = 8$ the two expressions above differ for a small term of order $|D(x) \cap D(y)|/|D(x) \cup D(y)|^2$. The second formulation of this measure is usually called *Jaccard dissimilarity coefficient* or *index* in text retrieval publications, and is a very simple quantity commonly used in classification problems of various kind (see chapter 3).

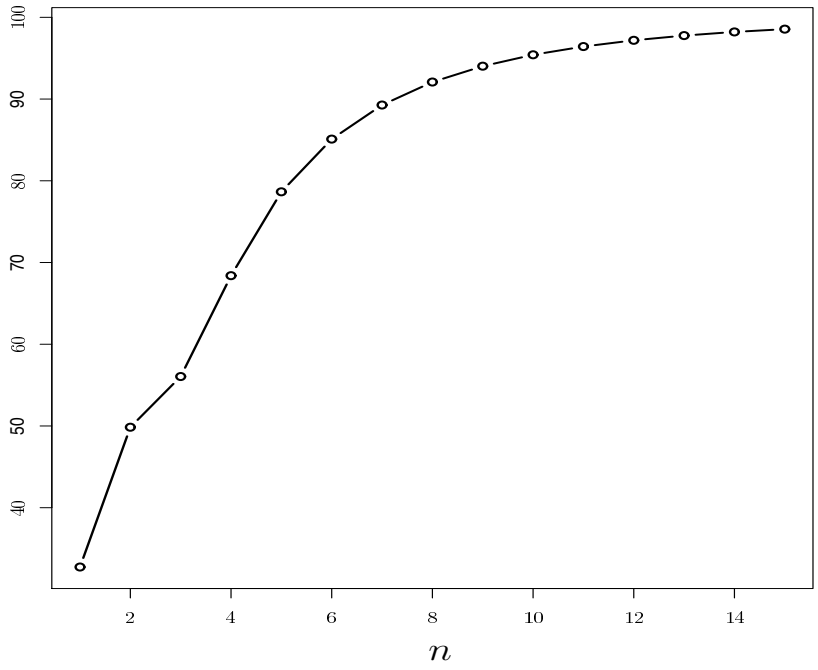


Figure 2.7: The mean fraction of the distance between two texts of our data set due to the n-grams which are present only in one of them.

Due to the previous considerations on n-gram statistics, it is not surprising that both these distances, with n large enough (e.g. $n \geq 7$), give the same results as the n-gram distance.

2.2.6 Ranking analysis

Quantitative methods for authorship attribution do not use any *a-priori* information on the stylistic differences between authors. On the other hand, an efficient quantitative method can provide *a-posteriori* information. In this section we discuss some preliminary theoretical considerations on this subject.

In order to delimitate the problem, we assume to have only two sources: G (e.g. the author Gramsci) and N (e.g. the other authors). Supposing it

is possible to consider G and N as ergodic sources, the values of the n-gram distance calculated between two arbitrary texts from the two sources have a probabilistic distribution depending only on the sources G and N themselves. Nevertheless, a statistical analysis of these values would not be very useful, mostly because the distance between two texts strongly depends on their lengths: to put it differently, it seems unadvised to compare the distances between different pairs of texts, while it is much more sensible to compare the distance of two texts from a fixed one. Indeed, we are trying to define quantities which are invariant under a suitable class of transformations.

Given a text x from the source G, we consider the empirical distributions, denoted by g and n, of the distance between x and a random text of G and of N, respectively. With a monotonic transformation we can consider g as uniformly distributed in $[0, 1]$ and n as a continuous variable taking values in $[0, 1]$. We denote by m_n the distribution function of n with respect to g: if $z \in [0, 1]$ is the fraction of G-texts with distance less then a fixed value d from the text x, then $m_n(z)$ is the fraction of N-texts whose distance from x is less then d.

We will recover m_n from the data on the distances of the training set. Once we have fixed $x \in G$, we have the values of the distance between x and 49 other Gramscian texts, which we indicate with $g_1 < \ldots < g_{49}$. We also have the 50 values $n_1 < \ldots < n_{50}$ of the distances between x and the 50 non-Gramscian texts. An empirical approximation of m_n is given by the following formula: let $k \in \{1, \ldots 49\}$,

$$m_n\left(\frac{k}{50}\right) = \frac{1}{50}\sum_{i=1}^{50} \mathcal{X}\{n_i \leq g_k\},$$

where $\mathcal{X}$ is the characteristic function. We also set $m_n(0) = 0$ and $m_n(1) = 1$. The function $m_n(z)$ is the non-Gramscian mass which corresponds to the Gramscian mass z. Taking the average for x varying in all the Gramscian texts G, we obtain the function with the plot on the left of figure 2.8 (linearly interpolated in the intervals $[k/50, (k + 1)/50]$).

In the same way we obtain the distribution function m_g (plot on the right

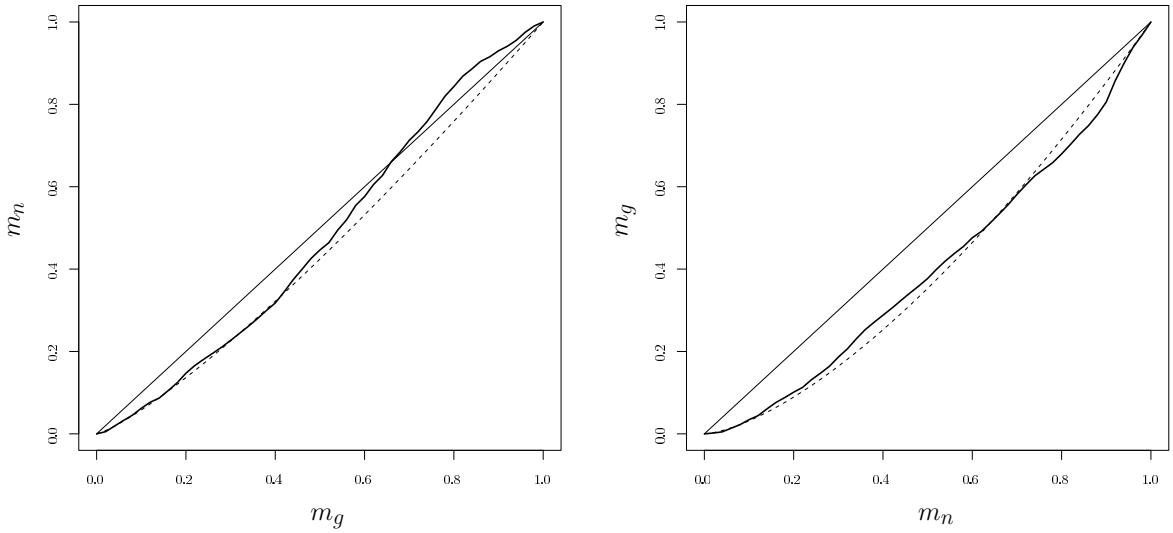


Figure 2.8: The distribution function m_n with respect to m_g for Gramscian texts, and the distribution function m_g with respect to m_n for non-Gramscian texts. Dashed lines are the power law approximations of the two functions.

of figure 2.8), which is the distribution function of the distance of the Gramscian texts from a given non-Gramscian text, with respect to the distribution of the distances of non-Gramscian texts.

Observing the two graphics we can make some considerations.

- In both cases, in the initial part the "mass" for the "wrong" author is less then the mass for the right one, according to the fact that the n-gram distance is a good indicator for authorship.

- The initial part of both curves can be approximated by a power law. A log-log regression (without the constant term) gives the exponent 1.237 for m_n and the exponent 1.501 for m_g.

- The distribution function m_n changes its convexity for $m_g \approx 0.65$, and becomes larger than m_g. This means that, given a Gramscian text, the more distant non-Gramscian texts are closer to it than the farther

Gramscian texts. This statistical fact may show that Gramsci was a more various author then the others (note furthermore that the author N is indeed a collection of 17 different authors).

This analysis suggests a way to define theoretically a Gramscianity index, in the spirit of section 2.2.2. Consider now a text x of an unknown author (either G or N) and its distances from ℓ Gramscian texts and ℓ non-Gramscian texts. We indicate with $c \in \{G, N\}^{2\ell}$ the ranking of the authors of the references texts; for example, a ranking $c = (G, G, N, G, N, N)$ for $\ell = 3$ means that, sorting the 6 reference texts by growing distance from x, the first, second and fourth neighbours are Gramscian, while the third, fifth and sixth are non-Gramscian.

Assuming a given expression for m_n, we can calculate $P(c|\, x \in G)$, i.e., the conditional probability of observing the positions of G- and N-texts in c, if x is indeed a text by G:

$$P(c|\, x \in G) = \int_0^1 \rho_1(z_1)\partial z_1 \int_{z_1}^1 \rho_2(z_2)\partial z_2 \int_{z_2}^1 \rho_3(z_3)\partial z_3 \cdots \int_{z_{2\ell-1}}^1 \rho_{2\ell}(z_{2\ell})\partial z_{2\ell}, \tag{2.4}$$

where

$$\rho_i(z) = \begin{cases} 1 & \text{if} \quad c_i = G \\ m_n'(z) & \text{if} \quad c_i = N \end{cases}.$$

In a similar way we can express $P(c|\, x \in N)$ assuming a given expression for m_g.

Using Bayes formula, we can also calculate $P(x \in G|\, c)$ and $P(x \in N|\, c)$. We can then attribute x to G or to N by evaluating the sign of the difference $P(x \in G|\, c) - P(x \in N|\, c)$. Namely, we can identify $P(x \in G|\, c)$ with a Gramscianity index and $P(x \in N|\, c)$ with a non-Gramscianity index for the text x, and we can proceed as in section 2.2.2. In order to use Bayes formula we need to know the *a-priori* values of $P(x \in G)$ and $P(x \in N)$. We have used an equal number of texts of G and of N, so that $P(x \in G) = P(x \in N) = 1/2$. Therefore:

$$P(c) = \frac{P(c|\, x \in G) + P(c|\, x \in N)}{2},$$

and

$$P(x \in G|\, c) = \frac{P(c|\, x \in G)}{P(c|\, x \in G) + P(c|\, x \in N)}$$
$$P(x \in N|\, c) = \frac{P(c|\, x \in N)}{P(c|\, x \in G) + P(c|\, x \in N)}$$

The construction of these indices depends on the explicit calculation of $P(c|\, x \in G)$ and $P(c|\, x \in N)$, as in (2.4), which is easy for some particular choice of the laws $\rho_i(z)$. This is the case for a power law. In particular, if $m_n(z) = z^{1+\alpha}$, then

$$\rho_i(z) = (1+\alpha_i) z^{(1+\alpha_i)-1}, \quad \text{where } \alpha_i = \begin{cases} 0 & \text{if} \quad c_i = G \\ \alpha & \text{if} \quad c_i = N \end{cases}.$$

Inserting this expression in (2.4), we can now use the fact that, by direct integration,

$$\int_0^1 z_1^{\alpha_1 - 1} \partial z_1 \int_{z_1}^1 z_2^{\alpha_2 - 1} \partial z_2 \cdots \int_{z_{k-1}}^1 z_k^{\alpha_k - 1} \partial z_k = \frac{1}{\alpha_1(\alpha_1+\alpha_2)\cdots(\alpha_1+\alpha_2+\ldots+\alpha_k)}. \tag{2.5}$$

Defining

$$\bar{m}_g(k) = \sum_{i=1}^k \mathcal{X}\{c_i = G\}, \quad \bar{m}_n(k) = \sum_{i=1}^k \mathcal{X}\{c_i = N\}$$

we have $\bar{m}_g(k) + \bar{m}_n(k) = k$ and

$$\alpha_1 + \ldots + \alpha_k = k + \alpha \bar{m}_n(k) = k\left(1 + \alpha \frac{\bar{m}_n(k)}{k}\right).$$

Finally

$$P(c|\, x \in G) = \frac{(1+\alpha)^\ell}{(2\ell)!} \frac{1}{(1+\alpha \bar{m}_n(1)/1)\cdots(1+\alpha \bar{m}_n(2\ell)/(2\ell))}. \tag{2.6}$$

In the same way, if $m_g(z) = z^{1+\beta}$, we obtain

$$P(c|\, x \in N) = \frac{(1+\beta)^\ell}{(2\ell)!} \frac{1}{(1+\beta \bar{m}_g(1)/1)\cdots(1+\beta \bar{m}_g(2\ell)/(2\ell))}. \tag{2.7}$$

A simple expression for $P(x \in G|\, c) - P(x \in N|\, c)$ can be obtained by considering the first order in α, β of these expressions. We obtain

$$\beta \left(\sum_{k=1}^{2\ell} \frac{\bar{m}_n(k)}{k} - \ell \right) - \alpha \left(\sum_{k=1}^{2\ell} \frac{\bar{m}_g(k)}{k} - \ell \right).$$

Since $\bar{m}_g(k) + \bar{m}_n(k) = k$,

$$\begin{aligned} P(x \in G|\, c) - P(x \in N|\, c) &= (\alpha + \beta) \left(\textstyle\sum_{k=1}^{2\ell} \frac{\bar{m}_g(k)}{k} - \ell \right) \\ &= \frac{\alpha+\beta}{2} \textstyle\sum_{k=1}^{2\ell} \left(\frac{\bar{m}_g(k)}{k} - \frac{\bar{m}_n(k)}{k} \right). \end{aligned}$$

We obtain the following condition for the choice of G as the author of the text x:

$$\sum_{k=1}^{2\ell} \left(\frac{\bar{m}_g(k)}{k} - \frac{\bar{m}_n(k)}{k} \right) > 0.$$

We remark that this condition is very similar to the one used for the experiments described in the preceding sections, which, in this notation, is

$$\sum_{k=1}^{2\ell} \left(\frac{\bar{m}_g(k)}{k} \mathcal{X}\{c_k = G\} - \frac{\bar{m}_n(k)}{k} \mathcal{X}\{c_k = N\} \right) > 0.$$

The results for the 100 training tests are very similar for the two methods.

2.2.7 A quick comparison with other methods

As discussed in 2.2.1, the Gramscian corpus has many interesting characteristics from the point of view of authorship attribution studies. In a first attempt to compare our methods with other techniques on a common test ground, we used the *JGAAP* software, developed by a group leaded by P. Juola, the organizer of the AAAC.

Unfortunately, the "Documentation" section of the software web page[4] is very basic, and lacks a precise description of the methods; on the other hand, the source code is fully available and can be downloaded and compiled freely. We used version 3.1 of the software.

[4] `http://server8.mathcomp.duq.edu/jgaap/w/index.php/Main_Page`

The *Event set* section of the program allows to choose among the following features:

- characters
- words
- word lengths
- syllables
- character $n-$grams with $n = 2, 3, 4$
- word $n-$grams with $n = 2, 3, 4$

The method of analysis can instead be chosen among the following ones:

- cross entropy
- histogram distance
- KS distance
- Levenshtein distance
- Camberra distance
- Manhattan distance
- LDA (linear discriminan analysis)
- SVM (support vector machine)
- Gaussian SVM

We tried all the possible combinations of features and methods on the blind test corpus of 100 reference texts and 40 test documents described above, obtaining the results reported in table 2.5. We use here a standard performance indicator for text retrieval, the *F-measure*, defined as follows:

$$F\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall},$$

where, if TP (respectively TN) is the number of true positives (negatives) and FP (respectively FN) is the number of false positives (negatives),

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}.$$

The F-measure is therefore the harmonic mean of precision and recall.

The comparison shows that, at least for the Gramscian corpus, both of our attribution techniques work better than all of those available in Juola's software; indeed, the $F-$measure on the blind test corpus is 0.947 for the 8-gram distance, and 0.909 for the BCL method.

	char.	words	word lgths	syll.	char. 2−gr.	char. 3−gr-	char. 4−gr.	word 2−gr.	word 3−gr.	word 4−gr.
cross entropy	0.714	0.731	0.717	0.694	0.698	0.723	0.696	0.642	0.605	0.485
histogram d.	0.485	0.698	0.667	0.533	0.818	0.650	0.634	0.655	0.667	0.667
KS d.	0.667	0.625	0.615	0.429	0.556	0.516	0.588	0.714	0.619	0.462
Levenshtein d.	0.579	0.625	0.583	0.698	0.634	0.810	0.615	0.679	0.643	0.655
Camberra d.	0.698	0.667	0.486	0.537	0.711	0.750	0.714	0.667	0.667	0.667
Manhattan d.	0.667	0.625	0.615	0.429	0.556	0.516	0.588	0.714	0.619	0.462
LDA	0.595	0.711	0.732	0.684	0.872	0.864	0.732	0.766	0.667	0.667
SVM	0.833	0.650	0.667	0.703	0.833	0.878	0.833	0.745	0.667	0.667
Gaussian SVM	0.765	0.727	0.524	0. 636	0.842	0.864	0.776	0.667	0.667	0.667

Table 2.5: $F-$measure on the Gramscian blind test data set with the methods available in the JGAAP software. Features on the columns, attribution methods on the rows. Note that methods using character $n-$grams and machine learning techniques are the majority among those that have an $F-$measure higher than 0.8 (highlighted in red and underlined).

We do not pretend this comparison to be highly significant, since we merely used the JGAAP package as a "black box" by feeding it with the texts and accepting the results it returned, without being able to tune the methods by setting the parameters etc. Anyway, we can think to this test as a further indication of the good behaviour of our methods for authorship attribution. As a future work, we plan to do a more extensive and better grounded comparison on a quite large benchmark, in the spirit of the one that Grieve reported and discussed in [23].

2.3 Xenophon and Thucydides

2.3.1 The problem

When the Gramsci project was already under way, we started a preliminary collaboration with Luciano Canfora, a well known classical philologist from the University of Bari, Italy. He proposed to our group an interesting and demanding open problem of disputed authorship in Greek historical literature.

Canfora's theory [12, in Italian] is the following: in the traditional, Hellenistic partition of the *History of the Peloponnesian War*, the monumental work by Thucydides[5], the part that narrates the 22nd-27th years of the war between Athens and Sparta was erroneously attributed to Xenophon[6]. The first and a part of the second book of Xenophon's *Hellenica* should therefore be attributed to Thucydides instead, according to this thesis.

The problem is undoubtedly interesting for a stylometric study, and very different from the previous work on Gramscian articles. Apart for the obvious difference in the language, which will be further discussed in the next paragraph, an important element of distinction is the text length: even after removing a lot of characters to "normalize" the texts of the corpus, they are on average 50,000-100,000 character long, allowing for much richer statistics for any feature extraction method.

Unfortunately, the only exceptionally short texts in the corpus are precisely the first and the initial part of the second book of the *Hellenica*, that is, the disputed texts under analysis. Once again, when dealing with real-world problems there is no hope of having the "perfect corpus", and this is true even more for this Greek data set; a broader discussion on this subject

[5]Thucydides (c. 460 BC – c. 395 BC) was a Greek historian, and the first who wrote a detailed and scientifically built historical work in Greek's literature.

[6]Xenophon (c. 430 – 354 BC) was a Greek historian, soldier and mercenary, who wrote a number of historical works, among which the *Hellenica*, which is considered as the natural continuation of Thucydides' *History of the Peloponnesian War*, since it starts from where Thucydides had ended his narration.

will follow.

2.3.2 A first experiment

The textual source that we used to compose the corpus was the digital library of the *Perseus Project* of Tufts University[7]; we had to pretreat the texts to make them homogeneous, by removing HTML tags and all the characters that didn't belong to the Greek alphabet at the time when Thucydides and Xenophon wrote: lowercase letters and punctuation, indeed, appeared later in Greek writing, as well as most diacritical marks (accents, breathings, iota subscript).

The following operations were performed, in summary:

- subdivision into books, with the exceptions discussed below;
- deletion of HTML tags;
- reduction of the alphabet to only the letters and the whitespace separator;
- whitespace normalization;
- elimination of all diacritical marks;
- uppercasing.

The resulting alphabet is composed of only the 24 characters of the classical Greek alphabet, plus the whitespace.

The first corpus we used for our experiments was made up of the following texts, that we kept as a reference for the attribution:

Thucydides : the eight books of the *History of the Peloponnesian War*, with the chapters from 1 to 83 of book III excluded, since they are not signed by Thucydides himself (cf. again [12]);

Xenophon : the seven books of the *Anabasis*, the books from 3rd to 7th of the *Hellenica* and the second part of the second book of that same

[7] `http://www.perseus.tufts.edu/`

work (from 3.11 to the end, the part that was certainly written by Xenophon).

The texts are quite long, as already discussed: their lengths are shown in figure 2.9.

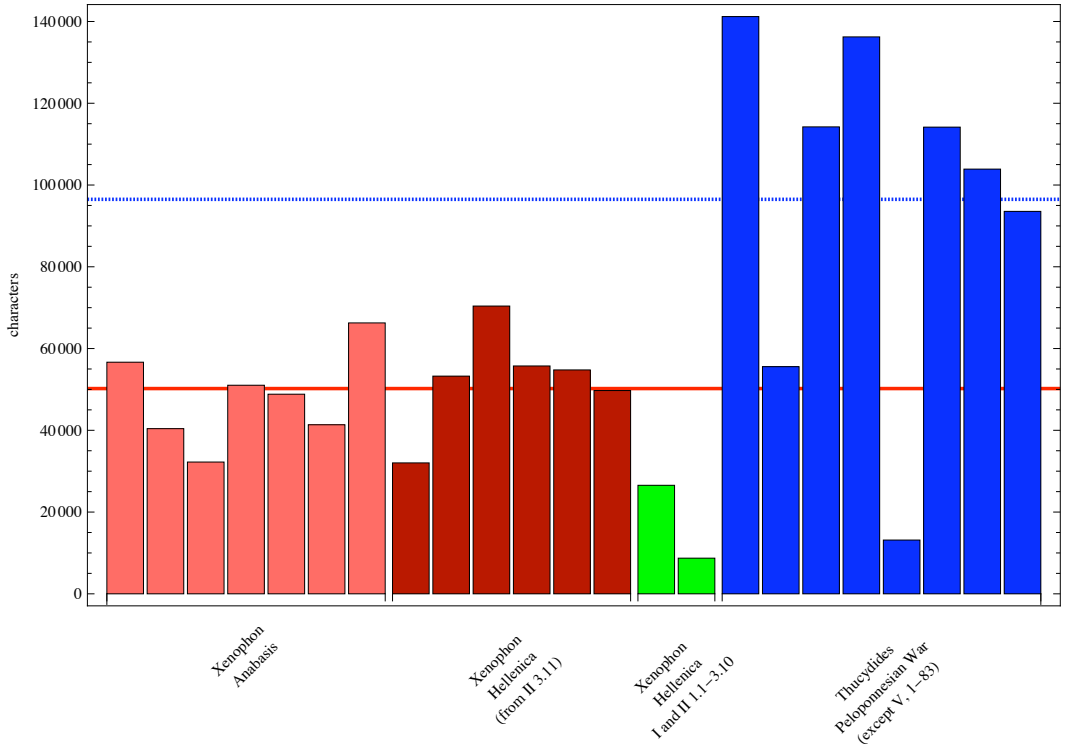


Figure 2.9: Lengths of texts by Xenophon and Thucydides, together with their averages (solid and dashed horizontal lines, respectively). Note that the two disputed texts, shown in green, are among the shortest of the whole corpus.

In order to have a cross-validation of the results, we used the same texts also as a test set, together with the two texts for which we were trying to determine the attribution, namely the first book and the second book up to 3.10 of the *Hellenica*.

We tested on this corpus exactly the same $n-$gram distance d_n that we had succesfully used for the Gramsci project; since the right value of n depends strongly on the language (and possibly on many other characteristics

of the corpus), though, we didn't fix a single value of the parameter, which was allowed to vary between 1 and 15.

The results of this first experiment were surprisingly good, possibly even *too* good: for any value of $n \geq 2$ the reference texts are correctly parted between the two authors. This result holds already when we consider only the first neighbour, and it is confirmed when we use a weighted index like the one defined in paragraph 2.2.2. The two disputed texts are attributed in any case to Xenophon.

2.3.3 Are simple statistics enough?

Considering the extraordinary good classification we had obtained in the first experiment, we needed to understand if there was some very basic difference between the styles of the two authors, something that could be detected by extracting very simple statistics from the corpus.

The character distribution, shown in figure 2.10, doesn't seem to highlight any notable difference between the two authors. This is indeed consistent with the fact that $n = 1$ is the only value of the parameter for which the distance gives the wrong attribution for some texts.

The word length distribution is shown in figure 2.11. It is interesting to note the difference in the distribution for words between 4 and 10 character long, for Xenophon and Thucydides. More interestingly, the distribution for the disputed texts (green squares) is nearer to the one for Thucydides.

2.3.4 Extending the corpus

A sligthly deeper analysis of the results of the first experiment shows that each text has as its first neighbour not only a text written by the same author, but always another chapter of the same book from where the text itself was extracted. The suspicion that the attribution was based on the subject, instead of the authorship, was therefore very strong.

We tried to avoid as much as possible this risk by expanding the data set

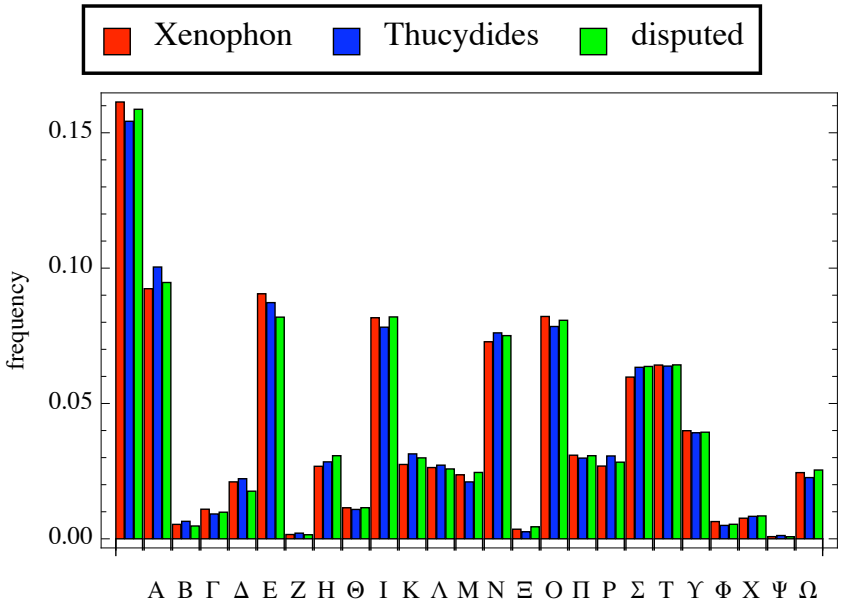


Figure 2.10: Distribution of the character of the alphabet in the two authors' reference texts, compared with the one of the disputed texts (green).

in two directions:

1. either by removing the *Hellenica* from the reference corpus and adding in its place chapters of other works by Xenophon (namely the eight books of the *Cyropaedia*[8]), dealing with other subjects;

2. or by adding texts written by other authors on the same subject: books XI, XII e XIII of the *Bibliotheca Historica*[9] by Diodorus Siculus and the Lives of Alcibiades, Lysander, Nicias and Pericles, from Plutarch's *Parallel Lives*[10].

[8]The *Cyropaedia* is a partly fictional biography of Cyrus the Great written by Xenophon in the early 4th century BC.

[9]The *Bibliotheca Historica* is a monumental work of universal history by Diodorus Siculus (90-27 aC).

[10]The *Parallel Lives*, or *Lives of the Noble Greeks and Romans*, is a work by Plutarch (46-127 dC) containing a series of biographies of famour men, arranged in pairs in order to underline common characteristics.

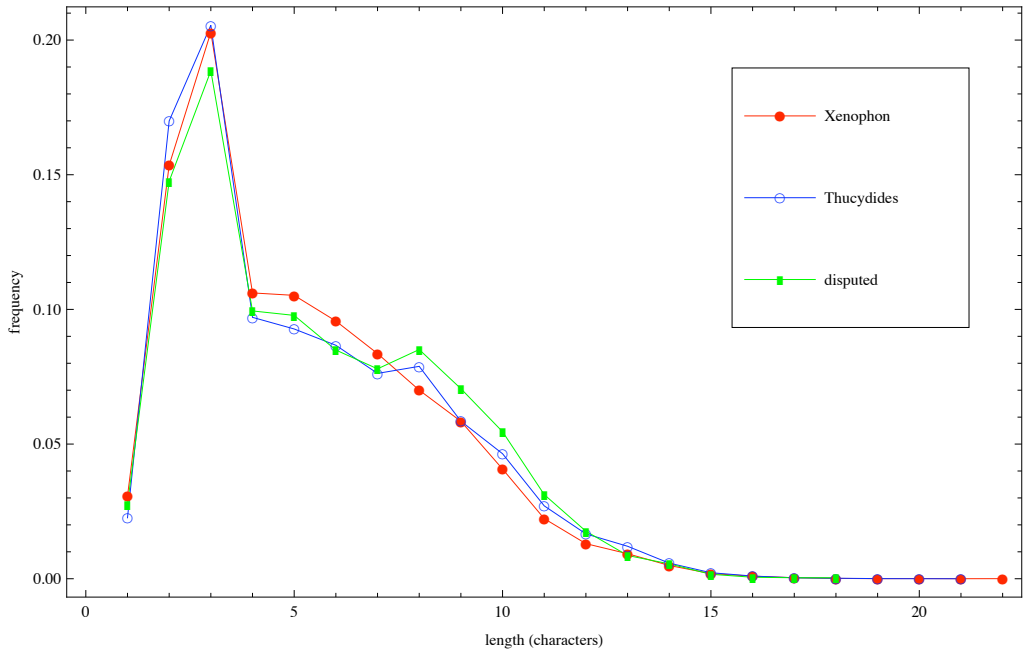


Figure 2.11: Word length distribution in the reference texts of the two authors, compared with the one of the disputed texts (green squares).

We want to underline that neither of these choices results in the *perfect corpus*: in the first case, we weren't able to enrich the corpus with other works by Thucydides, who wrote only the *History of the Peloponnesian War*, whereas Xenophon was a much more prolific author. In the second option the works we added are very well centered on the same subject, but from very different ages then the ones of the original corpus: as we have already discussed, the time when a certain work was written is a key factor in determining the style, and comparing texts separated by a time span of three or four centuries is not a very sensible choice. We want to underline that this is precisely the kind of problems that one is faced with when dealing with real-world attribution corpora; this is why we decided to try our methods on these new data sets anyway, even if with the awareness that we were dealing somehow with "borderline" data.

With the first of the two corpora described above we obtained results

perfectly comparable with those we got with the original corpus, i.e., all the texts of known authorship were correctly attributed; in this case, though, the value of n for which this happens is higher: $n \geq 7$. Considering only the first neighbour, the two disputed texts are here attributed to Thucydides, while they are assigned back to Xenophon with a weighted index on a larger number of neighbours. It is interesting to note that the first chapter of the *Hellenica* has the eigth book of the *Peloponnesian War* as its first neighbour, for any value of n, a fact that suggests a very high resemblance between the two texts.

The second proposed corpus gave for the first book of the *Hellenica* exactly the same results as the experiments described up to now, i.e., the attribution depends on whether or not the other chapters of the *Hellenica* are among the reference texts. The texts by Diodorus and by Plutarch are in general correctly assigned to the corresponding authors, with the exception of the XII book of Diodorus' *Bibliotheca Historica*, which goes to Thucydides for any value of the parameter: this could indeed be a case where the subject is more relevant than the author's style for the attribution. The first part of the second book of Xenophon's *Hellenica*, instead, has as its first neighbour the XIII book of Diodorus' *Bibliotheca Historica*; we want to underline once again, though, that this text is very short with respect to all those that compose the corpus, and therefore its attribution is certainly less sure than the one of the other works.

The preliminary experiments we performed on the same corpora by using our compression-based method described in section 2.2.3yielded essentially the same results.

In conclusion, the results we obtained for the Greek corpora are certainly very good from the point of view of the attribution of reference texts, but they do not allow us to state whether these attribution results depend on the subject or on the author's style; hence, we cannot use them to give a convincing attribution for the disputed texts.

It is certainly true, though, that even simple statistics indicate a simi-

larity between the beginning of the *Hellenica* and Thucydides' work, which is confirmed by observing that the eigth book of the *Peloponnesian War* is often in the first positions of the distance rankings for the disputed texts, for various values of the $n-$gram distance. Note that the VIII book of the *Peloponnesian War* has been the object of quantitative authorship verification studies itself, among which [36, in Italian] by M. Lana, who suggested a comparison with Xenophon's work as the next step of his research.

2.4 Conclusions and future developments

Authorship attribution is a very wide area of research, where a huge variety of methods have been experimented. The relatively recent interest in non-linguistic methods is certainly confirmed by the results of our experiments with $n-$gram statistics and data compressors, however limited such experiments may be. For sure, indeed, they give positive indications about the possibility of adopting those methods for the study of authorship; and with the Gramsci project we had the occasion of testing them on a real-world and at the same time well organized (in the sense discussed above) and controlled corpus.

What emerges in the tests with the $n-$gram distance is also the good performance of relatively large values of the parameter n, like 7, 8 or 9. The question about which value to choose for n is indeed completely open, also in the literature. If it is true that a "large" n is more exposed to the risk of catching the subject of the text (cf. [23, 62]), since it takes into consideration long, content-related words, on the other side such large values outperformed smaller ones both in our experiment on the Gramscian corpus, whose homogenity by topic has already been discussed, and a in controlled test that Clement and Sharp [14] performed using movie reviews extracted from the *Internet Movie Database*[11]. The reason for this fact is not clear, and deserves more attention. A possible interpretation is that a "large" n

[11] `http://www.imdb.com/`

is able to account for more factors among those that determine the text; it behaves somehow like a "multiple" feature, that contains information about the distribution of punctuation, of syllables, of function and content words, of couples of words, etc. All of these informations, that are marks of the author's style, come in a single "package", with no possibility of being isolated from one another, with scarcely controllable but fruitful effects on the attributions.

It would certainly be worth to test our methods on a larger controlled corpus, and to compare it with other attribution techniques, in the spirit of [23]. With respect to Grieve's experiment, though, we would be interested in trying a broader comparison, by applying also more standard techniques like Principal Component Analysis (PCA) or machine learning methods.

For what concerns real-world attribution problems, we could certainly improve the Greek corpus, and we are thinking of possible new corpora too; a potential idea is to try and give indications about the authorship of the *Historia Augusta*, a collection of biographies of Roman emperors and usurpers who lived in the II-III century BC. The work is signed by six different authors, and dated around the IV century; however, there is an open controversy regarding both the dating and the authorship of the text, since various philological studies indicate it as a later work of a single author who would have simulated different styles. Stylometric studies have already been performed on this work, with alternate results (see for example the publications in [40]); we plan to test on it our attribution methods.

The attribution problems we have dealt with up to now have a common characteristic: the distinction is always between two authors, or an author and all the other ones, considered as a whole, as in the Gramscian corpus. This two-class problem is realistic but also limited: one can easily imagine situations where a text is disputed among three or more authors. We would like to find suitable ways of extending our approach to this more general situation, as well as to adapt our ranking technique to the case when the classes are not homogeneous by the number of texts.

Chapter 3

Applications: plagiarism

Sheldon: You two may want to know that when I publish my findings, rest assured that your contributions will not go forgotten.
Howard: Thanks.
Sheldon: Although I will not have time to mention you on my Nobel speech, when I get around to writing my memoirs, I promise you a very effusive footnote, and maybe a signed copy.
- *The Big Bang Theory*, season 3, episode 1 -

3.1 Plagiarism detection

The phenomenon of plagiarism has been seeing a growing interest in the last two decades, with the rapid development of the world wide web, a tremendously vast source of documentation on virtually any subject of the human knowledge. We want to concentrate here on plagiarism for written texts, i.e., the re-use of (parts of) someone else's writing or written ideas without a proper citation or reference; this can happen either intentionally or unintentionally. There are various mechanisms that can be put in place to disguise an act of plagiarism, among which rephrasing the copied texts, changing the structure of the sentences, using synonims, and so on. Therefore, it is not enough for an algorithm for automatic plagiarism detection to identify identical passages: a more refined idea is needed. Furthermore,

some cases of more "borderline" plagiarism, like the re-use of someone's idea without a direct copying of any of his writings, can make the task even more difficult, if not impossible, for automatic methods.

As P. Clough underlines in his reviews of existing methods and softwares for plagiarism detection [15, 16], the origin of such research lies in the analysis of program code authenticity. Programming languages, though, are much simpler to analyse than natural language, due to the presence of structure and the closed vocabulary: often a comparison of the structures of two programs can be enough to generate or confirm a suspicion of plagiarism. For natural language writing, instead, no fixed structure or vocabulary is given, and the research often disregards complex NLP[1] techniques, and is instead driven towards simpler, statistical methods, like those that we have seen applied to authorship recognition issues. The problem of plagiarism is indeed strictly related to the one of authorship attribution: what we are looking for is an information about the style of the text, i.e., those characteristics that mark it as the work of some precise writer. On the other hand, as Clough correctly observes, this problem is also different from authorship attribution, in that it also strongly content-related, and it cannot be reduced to an analysis of style.

In 2009 the Web Technology and Information Systems Group at the Bauhaus-Universität Weimar, in Germany, and the Natural Language Engineering Lab at the Universidad Politécnica de Valencia, Spain, launched the *1st International Competition on Plagiarism Detection* [54] in the context of the PAN'09 workshop, with the aim of testing and comparing various methods for plagiarism detection on a common and well structured corpus. Due to the difficulty of finding a freely available corpus of real-world cases of plagiarism – in general, authors do not declare frauds openly! – the organizers had to build cases of *artificial* plagiarism out of a large collection of texts they downloaded from the Project Gutenberg website[2]. In order to re-

[1] Natural Language Processing

[2] `http://www.gutenberg.org`

produce as much as possible the mechanisms that a human author could use to disguise a plagiarized text, they automatically *obfuscated* the copy-pasted passages by using one or more of the following mechanisms [54]:

Random Text Operations: random shuffling, removal or insertion of words or short phrases;

Semantic Word Variations: random replacement of words with their synonims, antonyms, hyponyms or hypernyms;

POS-preserving Word Shuffling: plagiarized sentences are tagged with parts-of-speech (POS), and then words are shuffled in a way that preserves the original POS sequence.

Even if the automatic obfuscation doesn't reproduce exactly the mechanisms of human plagiarism (for example, the sentences it produces are often nonsense and can have sequences of repeated punctuation marks and other such irregularities), the corpus that was proposed was certainly interesting for plagiarism detection experiments, and indeed a number of groups partecipated in the contest, which was parted into two cathegories, corresponding to different real-world situations:

External Plagiarism: both the corpus of "suspicious" texts and the one of the possible sources of plagiarism were given, and the task was to identify whether or not each suspicious text contained plagiarized passages and the correct sources and positions from where they had been extracted. This corresponds to the situation where the source is available, for example in cases of plagiarism among students of the same class, self-plagiarism in the work of an author, etc.;

Intrinsic Plagiarism: the sources were not given, and the plagiarized passages had to be identified uniquely with an analysis of the suspicious text itself. This is often the case for real-world situations, when the text is somehow suspicious of not being original, but there is no idea about the source of such plagiarism, or the database of candidate sources is

so large that no computation on it is possible (e.g., the World Wide Web).

Ten groups participated in the first cathegory, and only four in the second one, which was certainly tougher. The results reported in [54] show interesting good performances for methods based on character $n-$grams: both the winners of the External and Intrinsic Plagiarism competitions used $n-$gram based methods, even if in very different ways. We refer to their respective works [24] and [63] for a complete description of their approaches, and will now concentrate on our contribution to the External Plagiarism contest. We will then make a few quick comments on intrinsic plagiarism in section 3.4.

3.2 PAN'09 competition: our method for external plagiarism

3.2.1 A general scheme for plagiarism detection

One of the main differences between the typical authorship attribution problem and most plagiarism recognition situations is certainly the *size* of the reference corpus. Whereas it is common and realistic to deal with few texts for each author in an authorship recognition framework, the PAN'09 corpus counts (already in its training section) more than 7000 suspicious texts and the same number of potential sources. As a side note, it is interesting to observe that plagiarism recognition research is most often developed by researchers in the field of computer science and information retrieval than authorship attribution is, a fact which probably has both computational and historical reasons.

Due to the size of the corpora, a generic method for the detection of external plagiarism can be divided into three steps, according to [64]:

i) heuristic identification of potential source documents: for each suspicious text a small subset of source documents has to be identified for the second and deeper (computationally demanding) analysis;

ii) exhaustive comparison of texts: the texts in the identified couples are compared in order to find those fragments that could have been copied, and the corresponding passages in the sources;

iii) knowledge-based post-processing, to eliminate proper citations from the plagiarism candidate fragments.

Our method fits in this framework, apart for the third point, which we completely disregarded. We will describe it in the next paragraphs, after a brief description of the PAN‘09 corpus.

3.2.2 The PAN‘09 corpus of external plagiarism

The organizers of the competition provided a training corpus, composed of 7214 source documents and 7214 suspicious documents, each with an associated XML file containing the information about plagiarized passages. A first statistical analysis shows that text lengths vary from few hundreds to 2.5 million characters, and the total number of plagiarized passages is 37046. Moreover, exactly half of the suspicious texts contain no plagiarism and about 25% of the source documents are not used to plagiarize any suspicious document. The length of the plagiarized passages has a very peculiar distribution, see figure 3.1: there are no passages with length in the window 6000-12000 characters, and even for long texts there is no plagiarism longer than 30000 characters; this is probably due to the artificial character of this corpus. A remarkable fact is that about 13% of the plagiarized passages consist of *translated* plagiarism.

Similarly, the competition corpus is composed of 7215 source documents and 7214 suspicious documents (obviously without any associated XML file). The length statistics are very close to those for the training corpus, see figure 3.2.

The details about the calculation of the mesuares of performance for the competition can be found in [54]; essentially, the overall score is calculated as

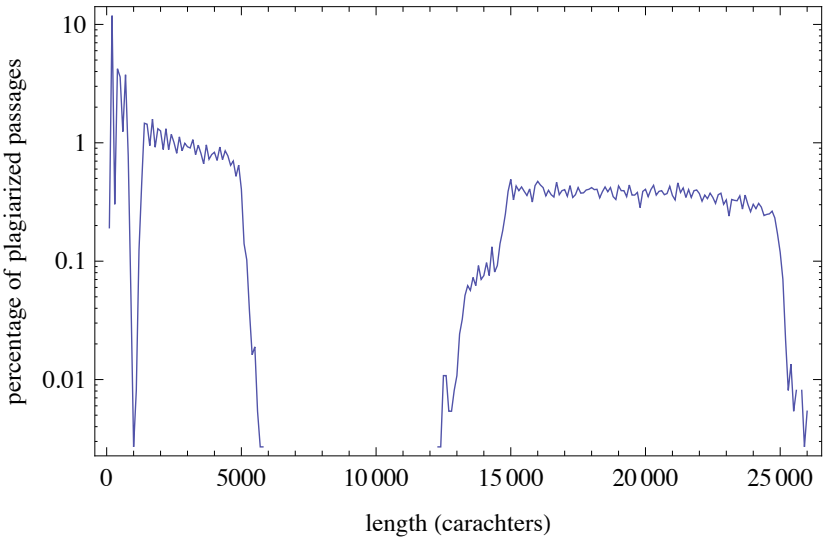


Figure 3.1: Distribution of plagiarized passage lengths in the training corpus.

the ratio between (character-level) F-measure and *granularity*, i.e., a measure of the average number of identified segments for the same plagiarized passage.

3.2.3 A first selection with word length coding

When we decided to partecipate in the competition, only one month was left for training, and we had to operate on a very tight time schedule.

Led by our experience in the field of authorship attribution, we decided to try to deal with plagiarism, at least for the first selection of relevant sources, as if it were a problem of style analysis; to this aim, hence, we applied the methods we had developed for authorship attribution, obviously adapting them to this context.

In order to identify, for each suspicious document, a set of candidate sources for plagiarism, we wanted to use the $n-$gram distance described in section 2.2.2, which had beed proved successful in authorship attribution problems. Since here we were looking for (obfuscated) copies of portions of texts, we expected a quite large value of n to give the best results[3]; com-

[3]Note indeed that the winning group used for this phase a kernel-based method with

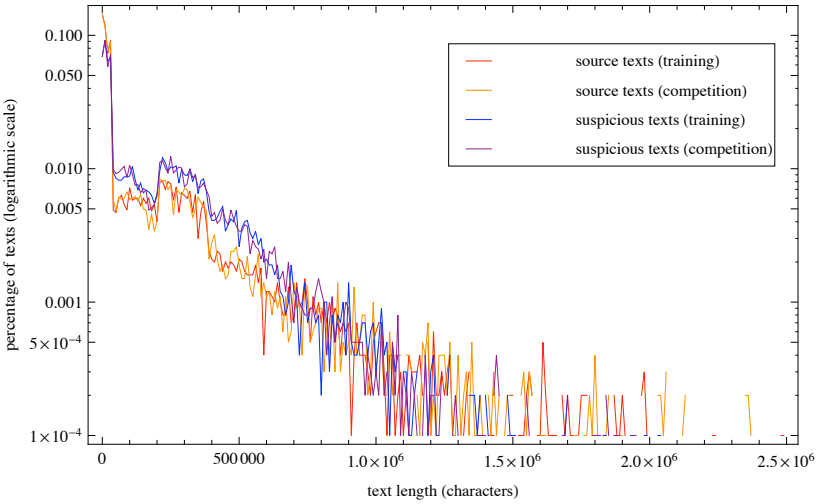


Figure 3.2: Text length distribution for the training corpus and for the competition corpus.

putation time requirements, though, made a comparison of the 7214×7214 texts impractical.

That is why we decided to apply a *coding*, similar to those described in section 1.2.1, but with the characteristic of being *lossy*: the texts were transformed into sequences of *word lengths*, so that for example the sentence `To be, or not to be: that is the question` becomes simply 2223224238. All word lengths greater than 9 were cut to 9, so that the new alphabet consisted of the nine symbols $\{1, \ldots, 9\}$. These encoded versions of the texts were on average 82.5% shorter than the original ones, and so computation times were greatly reduced.

Since it was impossible to do repeated computations of the $n-$gram distance with different values of n on the whole corpus, the tuning of the parameter was done by using a small subset of 160 suspicious texts and 300 source texts, suitably selected by imposing text and plagiarism length distributions comparable to those of the whole corpus. The parameter $n = 8$ was then chosen as a compromise between a good recall (the fraction of plagiarized

16-grams of characters, cf. [24].

characters coming from the first 10 nearest neighbours of each suspicious text is 81%) and acceptable computation times. Note that a recall of 81% is a very good result for the 8-gram distance, since the method just described has basically no hope of recognizing the 13% of translated plagiarism. Therefore, at least on the small subset of the training corpus, only about 6% of the (non-translated) plagiarized passages are lost in this phase.

For each suspicious text the first 10 source neighbours ordered according to this distance were kept for further analysis.

3.2.4 Detailed analysis: T9, matches and "squares"

After identifying the 10 most probable sources for each suspicious document, we performed a more detailed analysis to detect the plagiarized passages. Our (very simple) idea was to look for common subsequences (*matches*) longer than a fixed threshold. To this goal we needed to recover some of the information we lost in the first passage, by first rewriting the text in the original alphabet and then using again a coding, but a different, less "lossy" one, once again to reduce computational times and also in order to be able to apply a fast algorithm for the identification of longest matches, as it will be discussed below. We performed a *T9-like* coding, emulating the system that is used for assisted writing on most mobile phones. The idea is to translate three or four different letters into the same character, for example $\{\texttt{a,b,c}\} \mapsto 2$, $\{\texttt{d,e,f}\} \mapsto 3$ and so on, as in a typical mobile phone keyboard. The symbol 0 is used for newline and blank space, 1 for all symbols other than these two and the letters of the alphabet. The new alphabet for the encoded texts is therefore made up of 10 symbols: $\{0, 1, 2, \ldots, 9\}$. Note that the use of T9 "compression", which could seem strange at a first sight, can be justified by observing that a "long" T9 sequence (10-15 characters) has in most cases an "almost unique" translation into a sentence which makes sense in the original language: this is what allows a mobile phone user not to switch too often to words different from the one that the system suggests as the first guess, especially for long words. Also, in this way we got no

change in the length of texts, and we were able to avoid the use of indices to reconstruct the actual positions and lengths of matches.

The "true" matches between suspicious and source documents were then found: from any possible starting position in the suspicious document the longest match in the source document was calculated (possibly more than one with the same length). If the length was larger than a fixed threshold and the match was not a submatch of a previously detected one, it was stored in a list.

Here, we took advantage of the choice of the T9 coding, which uses ten symbols: for any starting position in the source document, the algorithm stores the last previous position of the same string of length 7, and for any possible string of length 7 it is memorized, in a vector of size 10^7, the last occurrence in the source file. With respect to other methods (suffix trees or sorting, for istance), in this way we can search the maximum match in the source document while avoiding to care for shorter ones.

The threshold for the match length was arbitrarily fixed to 15 for texts shorter than 500,000 characters, to 25 for longer texts.

This algorithm provided us with a long list of matches for each suspicious-source pair of documents. Since the plagiarized passages had undergone various levels of obfuscation, the matches were typically "close" to each other in the suspicious texts but taken from not necessarily subsequent places in the source texts. If we represent the pair of texts in a bidimensional plot (cf. also the *Dotplots* in [16]), with the suspicious text on the x axis and the source text on the y axis, each match of length l, starting at x in the suspicious document and at y in the source document, draws a line from (x, y) to $(x + l, y + l)$. The result is often something similar to figure 3.3: there are some random (short) matches all around the plane but there are places where matches accumulate, forming lines or something similar to a square. Non-obfuscated plagiarism corresponds to lines, i.e., a single long match or many short close matches which are in succession both in the suspicious and in the source texts, whereas intuitively obfuscated plagiarism corresponds to

"squares": here the matching sequences are in a different order in the source and suspicious documents.

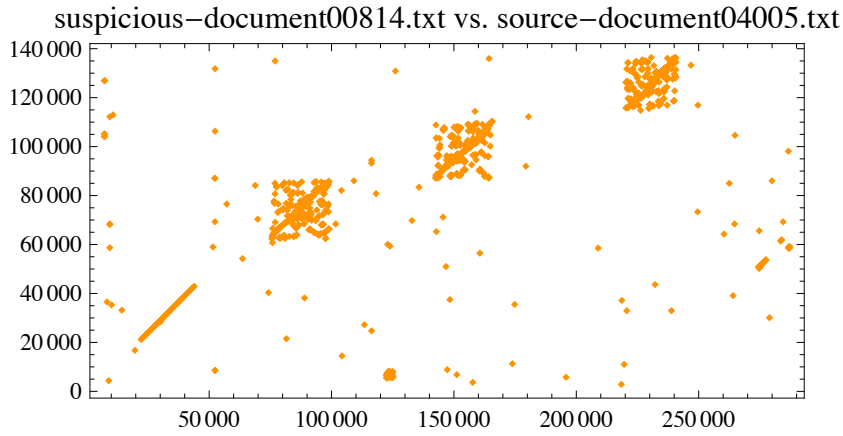


Figure 3.3: Orange points correspond to the position of matching characters (measured in number of chars from the beginning) between a suspicious and a source document of the training corpus. A "square" of matches corresponds to an obfuscated plagiarism.

Figure 3.4 is an example of what can happen when there is no plagiarism: the matches are uniformly spread around the plane and do not accumulate anywhere. Obviously these are just two of the many possible settings: longer texts or the presence of "noise" (e.g. long sequences of blanks, tables of numbers...) can give rise to a much higher density of matches, substantially increasing the difficulties in identifying the correct plagiarized passages.

In order to provide a single quadruple (x, y, l_x, l_y) of starting points and lengths for each detected plagiarized passage we needed to implement an algorithm that joined the "cloud" of matches of each "square". Note that the algorithm that performs this task needs to be scalable with plagiarism lengths, which can vary from few hundreds up to tens of thousands characters.

The algorithm we used here *joins two matches* if the following conditions hold simultaneously:

1. the matches are subsequent in the x coordinate;

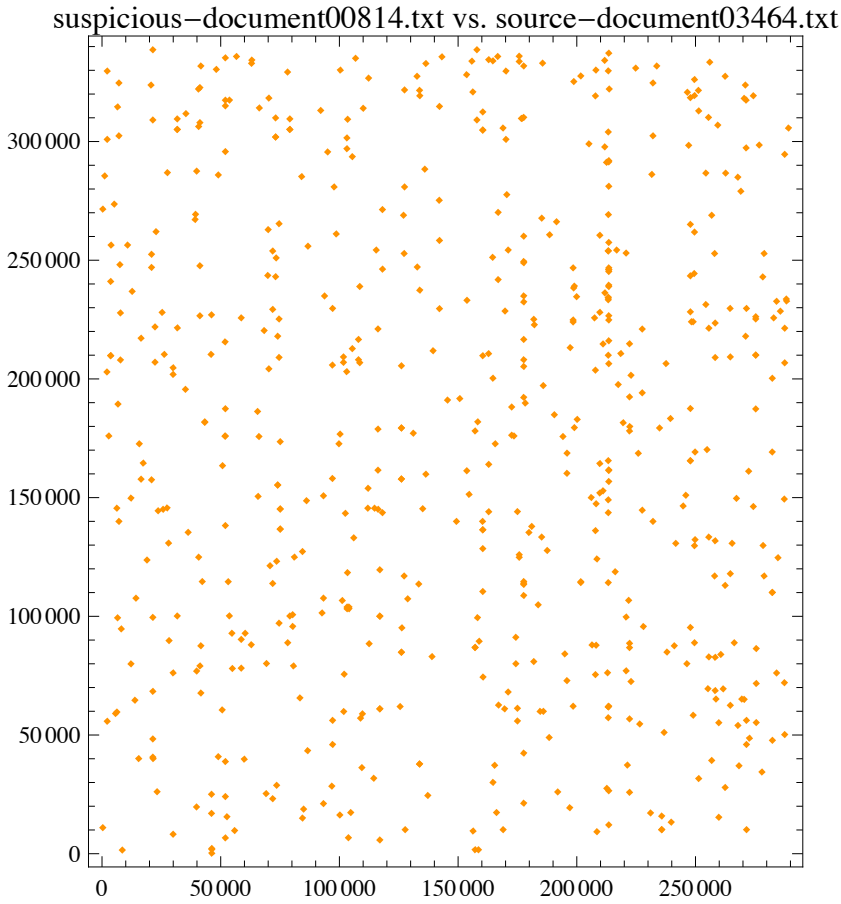


Figure 3.4: Orange lines correspond to the position of matching characters (measured in number of chars from the beginning) between a suspicious and a source document of the training corpus. No plagiarism is present in this case.

2. the distance between the projections of the matches on the x axis is greater than or equal to zero (no superimposed plagiarism) but shorter than or equal to the l_x of the longest of the two sequences, scaled by a certain δ_x;

3. the distance between the projection of the matches on the y axis is greater than or equal to zero but shorter than or equal to the l_y of the longer of the two sequences, scaled by a certain δ_y.

Merging repeatedly the segments which are superimposed either in x or

in y, we obtained some quadruples, which corresponded roughly to the "diagonals" of the "squares" in figure 3.3. To conclude, we ran the algorithm once again but using smaller parameters δ'_x and δ'_y, in order to reduce the granularity from 2 to approximately the optimal value of 1. Figure 3.5 shows the result of the algorithm for the couple of texts of figure 3.3 (blue), and figure 3.6 shows the very good superimposition with the actual plagiarized passages (black), as derived from the XML file.

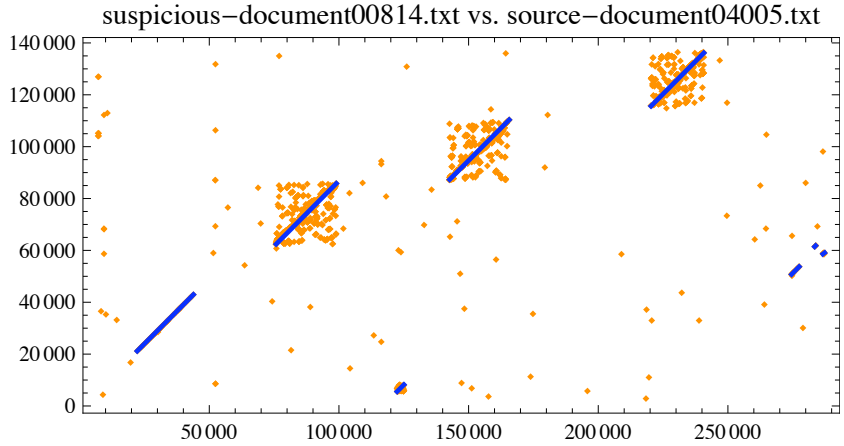


Figure 3.5: Detected plagiarism for the pair of texts of the training corpus indicated at the top of the plot. Single matches in orange, joined matches in blue.

Note that "joining algorithm" described above depends on 4 parameters: δ_x and δ_y for the first joining phase, and the rescaled δ'_x and δ'_y for the second joining phase. Our choice of the actual values in use was dictated essentially by the lack of time and no rigorous and efficient optimization was performed. Driven by very few trials and with some heuristics, we decided to use the following values: $\delta_x = \delta_y = 3$ and $\delta'_x = \delta'_y = 0.5$.

It is important to remark that different choices of the δ values yield to different detection results. For example, increasing their values typically results in a larger recall and in a better granularity, but also in a smaller precision. A further analysis of these dependencies could provide a controllable way of modifying the precision, the recall and the granularity, depending on the plagiarism detection task into consideration. A promising strategy that we plan

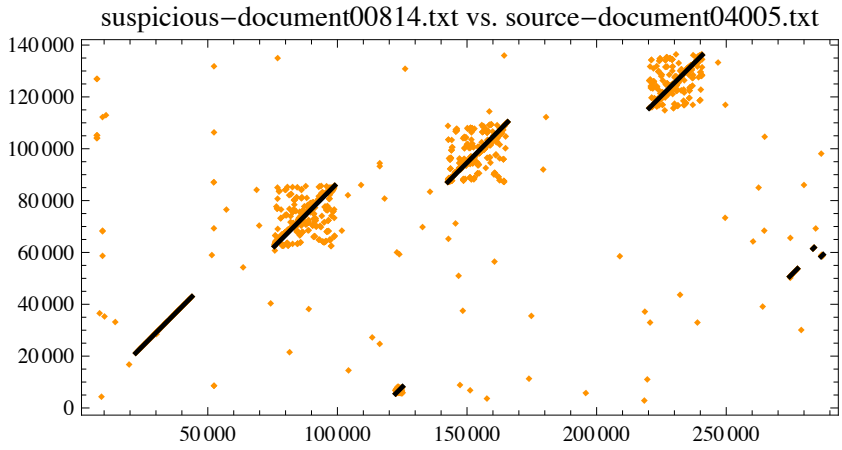


Figure 3.6: Plagiarized passages for the pair of texts of the training corpus indicated at the top of the plot. Single matches in orange, actual plagiarism in black. Note the perfect superimposition between the blue lines in figure 3.5 and the black lines here.

to explore in the future consists in a dynamical tuning of these parameters, according, for example, to the density of matches or to the lengths of the two texts into consideration.

3.2.5 Results and comments

The described algorithm gave the following results on the competition corpus [54]:

- Precision: 0.6727
- Recall: 0.6272
- F-measure: 0.6491
- Granularity: 1.1060
- Overall score: 0.6041

The overall score was the third best result after 0.6093 and 0.6957 of the first two partecipants. We stress that the overall scores dropped considerably starting from the fourth position (0.3045), the fifth (0.1885), and so on.

Moreover, while the winner had better results in all precision, recall and granularity, our precision was better than the one of the second partecipant, while recall and granularity were worse.

3.3 Back to the word length coding

How lossy is the word length coding precisely? And to what level encoded texts can be used as a valid alternative to the original ones? We had no time to address these questions with the due completeness during the PAN competition, but in a later study [2], done in collaboration with one of the groups that organized the contest, we estimated its efficiency in deeper detail.

An answer to the first question comes from the comparison between the distributions of word $n-$grams (sequences of n words, without coding) and word length $n-$grams, shown in figure 3.7 for a set composed of 500 documents extracted from the PAN corpus. Note that, as n grows, the two distributions tend to coincide, and a large superimposition is reached already for $n = 12$. This observation supports empirically the intuitive idea that, for a large enough n, the length coding is "almost injective", i.e., very few word $n-$grams are mapped to the same length $n-$gram.

In order to test the appropriateness of the length coding for the selection of potential sources for plagiarism, we also performed a few other experiments on the PAN training corpus. First we used a repeated sampling technique: for every run, we selected a small random subset of suspicious documents documents and an appropriate subset of reference documents, and evaluated the performance (cf. figure 3.8(a)).

Instead of the $n-$gram distance d_n, we used here an analogous of the simplified measure defined in section 2.2.5, the *Jaccard similarity coefficient* (cf. [27, in French]):

$$J_n(s,t) := \frac{|D_n(s) \cap D_n(t)|}{|D_n(s) \cup D_n(t)|}, \tag{3.1}$$

where s and t are two texts and, again, $D_n(s)$ is the $n-$gram dictionary

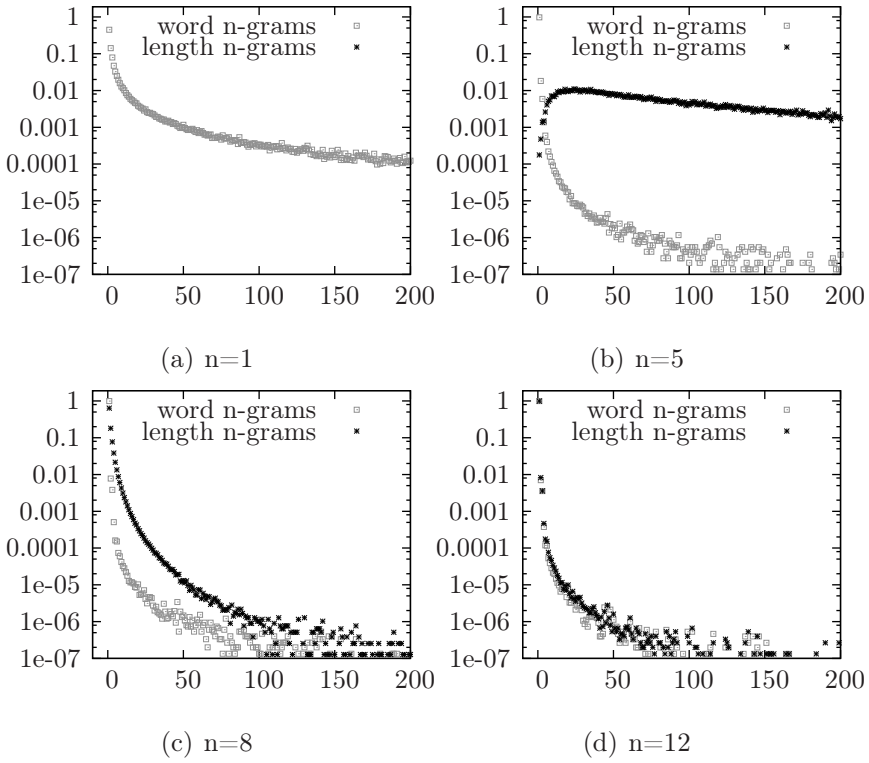
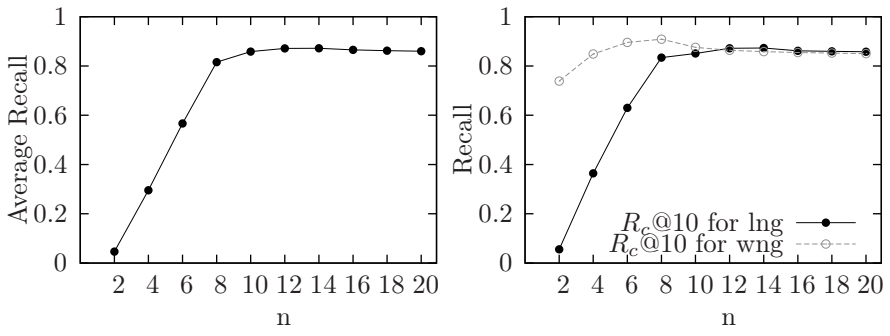


Figure 3.7: Frequency distributions for length $n-$grams (black stars) and for word $n-$grams (grey squares), for some values of n. The number of occurrences lies on the $x-$axis, with the corresponding percentage of n-grams on the $y-$axis. The length $n-$gram distribution converges to the one of word $n-$grams as n grows. No stars appear in the first plot because we show up to 200 occurrences only, which is lower than the frequency of any possible 1-gram of length encoded text in a representative corpus.

of s. The 10 nearest source texts for each suspicious document were then selected, exactly as it happened in the competition, and the performance was evaluated in terms of character-level recall over those 10 neighbours, R_c@10.

We also compared the recall with word length $n-$grams with the one we obtained with the standard word $n-$grams on the same subset of 160 suspicious and 300 reference texts used in the training stage of the competition; the results are shown in figure 3.8(b).

The obtained results confirmed an intuitive fact: there exists a threshold



(a) (b)

Figure 3.8: Character-level recall in the first 10 neighbours (R_c@10) for the PAN-PC-09 corpus (a) by averaging over 100 samples of 150 random query texts and around 300 reference documents, with length coding; and (b) compared to word $n-$grams on a fixed small subset of the training corpus.

for n, here around $n = 12$, above which the length coding and the word $n-$gram methods are equivalent; to be true, here the coding method performs always slightly better than word $n-$grams, for $n \geq 12$, but such small differences may be not reliable due to the fact that we are using a small subset of the corpus.

An important observation, indeed, is that identifying the relevant texts in such small samples of the corpus is much simpler, from a purely statistical viewpoint, than the "real" task of detecting few relevant texts for each suspicious document in the whole dataset of 7214 sources. At this point, thus, having identified 12 as a proper value for n, we calculated the Jaccard coefficient J_{12} on the whole PAN training corpus and obtained a recall R_c@10 = 0.86, a value even higher than the one shown in figure 3.8(a) for $n = 12$ with the small samples. Considering, again, that 13% of the plagiarism cases in the corpus are cross-language (cf. [54]) and the method we propose here has no hope of retrieving such cases, we consider a recall above 0.85 to be a very good result.

Furthermore, other experiments on a multi-language corpus of Wikipedia article revisions (not shown here, cf.[2]) confirmed the very good behaviour of

the word length coding for the pre-selection of suspicious documents in cases not only of (artificial) plagiarism, but also in a more general situation of text re-use. Even there, a quite high value of n (between 8 and 12) performed the best, and the results were always slightly better than those obtained (with much longer computation times!) with standard word $n-$grams.

3.4 Intrinsic plagiarism

The detection of intrinsic plagiarism is certainly a much tougher problem than the one of external plagiarism recognition. When the set of possible sources is not available, or it is too large to allow for an extensive comparison with the suspicious texts, the method is forced to work only on the text itself, and the only possible measure is one that detects a change in the *style* of the suspicious work. The resemblance with the case of authorship attribution is therefore more evident for this kind of problems, and indeed the winner of that cathegory of the PAN competition was E. Stamatatos [63], whom we already cited for his studies on authorship. And indeed, Stamatatos used a technique based on character 3−grams and distances: he considered a sliding window on the suspicious text, and compared the 3−gram statistics for the window and the whole text by using a slightly modified version of Kešelj's $n-$gram distance [30]. He then identified as suspicious those passages for which the distance between window and whole text was higher than a certain threshold, that depends on the variance of the sequence of distance values obtained while the window slides along the text.

Only four groups partecipated to the competition in this cathegory, probably both because of the shorter history of intrinsic plagiarism detection, if compared to external plagiarism recognition, and of the higher level of difficulty of the problem. Indeed Stamatatos' method, that had the best performance for this task, got an overall score of only 0.2462.

Another group of partecipants, L. Seaward and S. Matwin [58], applied instead a technique based on LZ compression of sequences of occurrences of

certain word classes in the texts, like nouns, verbs, prepositions, etc. Their results on the competition corpus, though, were quite poor, being even below the baseline performance level of considering every suspicious document as plagiarism-free (cf. [54]). In the recent past a group of English researchers [47] used the compression of appended texts as an indicator for plagiarism, in the spirit of [8], to deal with a specific plagiarism recognition problem in biomedical student reports.

We did not partecipate in this contest, but certainly it would be interesting to test the methods that we developed for the Gramsci project on this corpus. Stamatatos's $n-$gram approach is not far from our methods for authorship attribution, and it allowed him to obtain interesting results. In recent, very preliminar experiments, we had the idea of testing also our compression-based BCL method, by adapting it to this context: a window of w characters slides through the text in steps of s characters, and for each step we calculate the compression rate of that excerpt of length w by allowing the compressor to "learn" from the text deprived of the window and "shifted" so that it begins with the first character following the window. In other words, the window is considered as the test document, and the remaining text, made cyclic and rotated, is taken as the reference text, and then the cross-compression method described in section 2.2.3 is applied.

Once again, as it was for the external plagiarism competition, setting the parameters is a difficult and key aspect in the development of the method. Figure 3.9 shows the results for two texts in the PAN'09 corpus, with a window length $w(l) = \lfloor l/50 \rfloor$, l being the length of the text into consideration and $\lfloor x \rfloor$ the integer part of x, and step $s(l) = \lfloor l/500 \rfloor$. We are not at all sure that this is the best choice of the parameters, and we are planning to perform more extensive tests in the near future. However, these first results show that the method definitely has something to say in this context, and deserves further investigation.

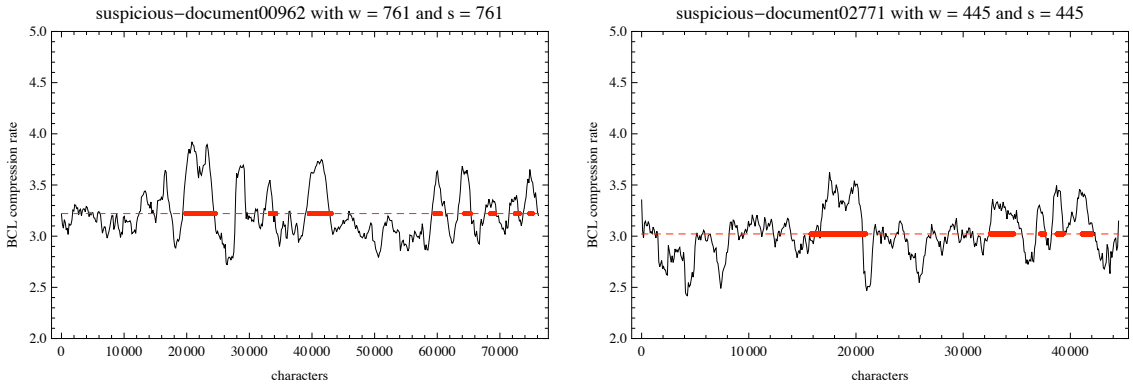


Figure 3.9: Results of the preliminary experiments on intrinsic plagiarism detection for two texts of the PAN'09 corpus. The plots show the rate of the cross compression (with BCL) of a window of length w that slides in steps of s characters, using the rest of the text as the database. The dashed horizontal line marks the average rate and the thick red segments indicate the plagiarized passages.

3.5 Concluding remarks

Plagiarism detection is in a sense a much more "technological" application of the ideas on information extraction from texts that we developed in chapter 1. However, it is certainly true that this problem has a great relevance in the field of information retrieval, and its importance will most probably grow with the spreading of the world wide web.

Apart for its practical interest, though, we believe it to be a very interesting test bench for our methods, and the presence of well structured textual corpora makes the comparison with other methods much simpler than in the case of authorship attribution. The results we obtained in the PAN'09 competition encouraged us to proceed in this direction, and the overall results of the different competitors certainly highlight the success of those methods that generally disregard a deep NLP approach in favour of simpler, non-syntactic approaches.

For intrinsic plagiarism, we plan to develop in the near future the recognition method sketched in the previous paragraph, that uses BCL compression, with a careful tuning of parameters. Also the $n-$gram approach deserves at-

tention here, as Stamatatos' first experiments underlined [63].

A further interesting possibility is to use a mixed approach, where intrinsic plagiarism techniques are used to detect stylistic difformities in suspicious texts, in order to recognize those that potentially contain plagiarism for a later comparison with possible reference texts.

The idea of using the word length coding to reduce the computational weight of $n-$gram extraction and comparison resulted in a quite successful method for the selection of source texts from a large database, that probably deserves further development. Note, furthermore, that the coding allows for the reduction of the cardinality of the alphabet to only 9 symbols; we started to analyse in [2] the possible advantages of this fact, like the ability of building in a single passage a prefix tree of the whole $n-$gram dictionary of the encoded text, so as to be able to use multiple values of n for the calculation of the distance.

Other interesting developments could come from automatic keyword extraction methods as the ones that we will discuss in the conclusions of this thesis: they could be used to select a restricted vocabulary of "interesting" words or $n-$grams on which a distance method could act.

For what concerns the "detailed analysis" step of external plagiarism detection, we can certainly improve greatly the approach that we used for the PAN competition. In particular, standard, less *ad hoc* clustering algorithm could be tested.

Discussion and perspectives

'Data! Data! Data!' he cried impatiently. 'I can't make bricks without clay.'
- A. Conan Doyle, *The Adventures of Sherlock Holmes* -

When dealing with applied problems as the ones we have presented in the two previous chapters, the temptation to jump from one real-world open problem to the other is always present. We have often let ourselves indulge in such temptation, and many of the experiments we tried in these years have never reached the needed level of accuracy to be included in a thesis.

As already discussed, though, at least for authorship attribution the literature and the number of available methods is so wide that probably the most necessary step in the research in this field at the moment would be a comparison on a well structured corpus, in the spirit of [23], but with a larger scope than the one that Grieve proposes in his work. Also, the field would certainly profit for a new competition, after the one that Juola proposed in 2003 [29]: this would force single groups to compare their methods on a common and hopefully solid ground. Indeed, the series of *International Competitions on Plagiarism Detection* [54, 55], to the first of which we participated, has proved to be a very stimulating environment for research and collaborations.

From the point of view of the applications, our interest is now focusing on the extraction of semantic information from a *single* text, in terms of keywords or automatic summaries. Some recent publications by M.A. Montemurro and D.H.Zanette [49, 50] propose an interesting idea for the ex-

traction of keywords from a written text through the analysis of the variation of entropy within the text itself. More precisely, their approach is to consider a word to be *important* for the text if its distribution (i.e., the distribution of its return times in the text, to use the terminology of chapter 1) is not uniform; such non-uniformity is measured by cutting the text into sections of fixed length and calculating the empirical entropy of the frequency distribution of words within the different sections, and then by comparing it with the one of a shuffled version of the text. We have experimented this approach on a number of texts, with decent results; we plan to investigate possible improvements, or combinations with other techniques, like the analysis of the representation of the text in terms of networks (graphs) of its words or $n-$grams, in the spirit of [44, 45].

The approach to this kind of problems could give us indications on a problem of classification of medical reports that *Noemalife S.p.A.* proposed to us recently. The documents are very short and noisy, which makes the problem really tough, and the goal is to assign to each of them one or more tags from a standard medical terminology, like the *ICD9* nomenclature[4], which is used by the Italian Ministry of Health. Some experiments with the *SNOMED* nomenclature[5] are available in literature, see for example [18, 53].

The theoretical framework from where our methods originated is the fruit of an interaction between various areas of research in mathematical physics and related fields (cf. [60]). We strongly believe in the power of such interaction, and would like to explore its potential in deeper detail. In particular, a very recent collaboration between our group and Benedetto and Caglioti at *La Sapienza* University led to a new estimator of relative entropy based on recursive pair substitutions [22, 7]. This and other estimators, like the one proposed in [11], deserve further analysis, in particular for what concerns their speed of convergence to relative entropy.

[4] `http://www.salute.gov.it/ricoveriOspedalieri/paginaMenuRicoveriOspedalieri.jsp?menu=classificazione`

[5] `http://www.ihtsdo.org/snomed-ct/`

Acknowledgements

This thesis is the fruit of the work of many hands and heads: it couldn't have existed without them.

I want to thank Dario Benedetto, Emanuale Caglioti, Maurizio Lana and Alberto Barrón-Cedeño for the fruitful and enjoyable discussions and collaborations.

Thanks to Sandro Graffi, who was the source of a number of potential problems of text classification, as well as ideas about how to deal with them.

A special thank goes to Martin Horvath, who wrote a part of the code I use to calculate the $n-$gram distance.

My colleagues Chiara Farinelli and Giampaolo Cristadoro have always supported my research, often by standing long hours of discussions on the details of a proof, sometimes simply by showing interest in my life and work.

Most of all I want to thank my advisor, Mirko Degli Esposti: for his generosity, his not always repaid belief in my capabilities, the enthusiasm in "dirtying his hands" with data and his passionate search for the best theoretical and practical approach to any scientific problem... or for a new, fascinating problem to deal with.

Last, I have to thank my family, who holds most of the responsibility for what I am an have now; my husband Andrea, who endured a painful period of agitation; and little Sam, who is coming on the tip of his toes, accepting my delirious rhythms without even interferring.

Bibliography

[1] A.R. BARRON, The strong ergodic theorem for densities: generalized Shannon-McMillan-Breiman theorem, *Annals of Probability*, **13** (1985), 1292-1303.

[2] A. BARRÓN-CEDEÑO, C. BASILE, M. DEGLI ESPOSTI and P. ROSSO, Word length $n-$grams for text re-use detection, *Lecture Notes in Computer Science*, **6008** (2010), 687-699.

[3] C. BASILE, D. BENEDETTO, E. CAGLIOTI and M. DEGLI ESPOSTI, An example of mathematical authorship attribution, *Journal of Mathematical Physics*, **49** (2008), 15211.

[4] C. BASILE, D. BENEDETTO, E. CAGLIOTI, G. CRISTADORO and M. DEGLI ESPOSTI, A plagiarism detection procedure in three steps: selection, matches and "squares", in *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09) (B. Stein, P. Rosso, E. Stamatatos, M. Koppel and E. Agirre, eds.)*, CEUR-WS.org, 2009, 19-23.

[5] C. BASILE, D. BENEDETTO, E. CAGLIOTI and M. DEGLI ESPOSTI, L'attribuzione dei testi gramsciani: metodi e modelli matematici, accepted for publication in *La Matematica nella Società e nella Cultura - Rivista dell'Unione Matematica Italiana*, 2010.

[6] C. BASILE and M. LANA, L'attribuzione di testi con metodi quantitativi: riconoscimento di testi gramsciani, *AIDAinformazioni*, **1-2** (2008), 165-183.

[7] D. BENEDETTO, E. CAGLIOTI, G. CRISTADORO and M. DEGLI ESPOSTI, Relative entropy via non-sequential recursive pair substitutions, *Journal of Statistical Mechanics: Theory and Experiments* (2010), P09010.

[8] D. BENEDETTO, E. CAGLIOTI and V. LORETO, Language trees and zipping, *Physical Review Letters*, **88** (2002), no. 4, 48702.

[9] W.R. BENNET, *Scientific and engineering problem-solving with the computer*, Prentice-Hall, Englewood Cliffs, NJ, 1976.

[10] M. BURROWS and D.H. WHEELER, *A block-sorting lossless dara compression algorithm*, Technical Report no. 124, Digital Systems Research Center, 1994.

[11] H. CAI, S.R. KULKARNI and S. VERDÚ, Universal divergence estimation for finite-alphabet sources, *IEEE Transaction on Information Theory*, **52** (2006), no.8, 3456-3475.

[12] L. CANFORA, *Il mistero Tucidide*, Adelphi, Milano, 1999.

[13] R. CILIBRASI and P. M. B. VITANYI, Clustering by compression, *IEEE Transaction on Information Theory*, **51** (2005), no. 4, 1523-1545.

[14] R. CLEMENT and D. SHARP, Ngram and Bayesian Classification of Documents for Topic and Authorship, *Literary and Linguistic Computing*, **18** (2003), no. 4, 423-447.

[15] P. CLOUGH, *Plagiarism in natural and programming languages: an overview of current tools and technologies*, Department of Computer Science, University of Sheffield, UK, Technical Report CS-00-05, 2000.

[16] P. CLOUGH, *Old and new challenges in automatic plagiarism detection*, National Plagiarism Advisory Service, 2003.

[17] T.M. COVER and J.A. THOMAS, *Elements of information theory*, Wiley Series in Telecommunications, Wiley, New York, 1991.

[18] L.M. DE BRUIJN, A. HASMAN and J.W. ARENDS, Supporting the classification of pathology reports: comparing two information retrieval methods, *Computer Methods and Programs in Biomedicine*, **62** (2000), no. 2, 109-113.

[19] A. DE MORGAN, Letter to Rev. Heald 18/08/1851, in *Memoirs of Augustus de Morgan by his wife Sophia Elizabeth de Morgan with selections from his letters* (S. E. De Morgan, ed.), Longman's Green and Co., London, 1851/1882.

[20] M. DEGLI ESPOSTI, C. FARINELLI, M. MANCA and A. TOLOMELLI, A similarity measure for biological signals: new applications to HRV analysis, *JP Journal of Biostatistics*, **1** (2007), no.1, 53-78.

[21] S. GALATOLO, Dimension and hitting time in rapidly mixing systems, *Mathematical Research Letters*, **14** (2007), no. 5, 797-805.

[22] P. GRASSBERGER, *Data compression and entropy estimates by non-sequential recursive pair substitution* , arXiv preprint `arXiv:physics/0207023`, 2002.

[23] J. GRIEVE, Quantitative authorship attribution: an evaluation of techniques, *Literary and Linguistic Computing*, **22** (2007), no. 3, 251-270.

[24] C. GROZEA, C. GEHL and M. POPESCU, ENCOPLOT: Pairwise sequence matching in linear time applied to plagiarism detection, in *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09) (B. Stein, P. Rosso, E. Stamatatos, M. Koppel and E. Agirre, eds.)*, CEUR-WS.org, 2009, 10-18.

[25] D.I. HOLMES, Authorship attribution, *Computers and the Humanities*, **28** (1994), 87-106.

[26] D.A. HUFFMAN, A method for the construction of minimum redundancy codes, *Proceedings of the IRE*, **40** (1952), 1098-1101.

[27] P. JACCARD, Étude comparative de la distribution florale dans une portion des Alpes et des Jura, *Bulletin del la Société Vaudoise des Sciences Naturelles*, **37** (1901), 547-579.

[28] P. JUOLA, Cross-entropy and linguistic typology, in *Proceedings of the Joint Conference on New Methods in Language Processing and Computational Language Learning (D.M.W. Powers ed.)*, Sydney, 1998, 141-149.

[29] P. JUOLA, Authorship attribution, *Foundations and Trends in Information Retrieval*, **1** (2006), no. 3, 233 -334.

[30] V. KEŠELJ, F. PENG, N. CERCONE and C. THOMAS, *N*-gram-based author profiles for authorship attribution, in *Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING'03 (V. Kešelj and T. Endo, eds.)*, Dalhousie University, Halifax, 2003, 255-264.

[31] V. KEŠELJ and N. CERCONE, CNG method with weighted voting in *Ad-hoc Authorship Attribution Contest (ALLC/ACH 2004)*, extended abstract, 2004.

[32] D.V. KHMELEV and F.J. TWEEDIE, Using Markov chains for identification of writers, *Literary and Linguistic Computing*, **16** (2001), no. 3, 299-307.

[33] I. KONTOYIANNIS, Asymptotic recurrence and waiting times for stationary processes, *Journal of Thoretical Probability*, **11** (1998), 795-811.

[34] I. KONTOYIANNIS, *Recurrence and waiting times in stationary processes, and their application in data compression*, PhD thesis, 1998.

[35] O.V. KUKUSHKINA, A.A. POLIKARPOV and D.V. KHMELEV, Opredeleniye avtorstva teksta s ispol'zovaniyem bukvennoi i grammaticheskoi informacii, *Problemy Peredachi Informatsii*, **37** (2000), no. 2, 96-108. Translated in Using literal and grammatical statistics for authorship attribution, *Problems of Information Transmission*, **37** (2001), 172-184.

[36] M. LANA, The authorship of VII book of Thucydides History. A study by correspondance analysis, in *Proceedings of Montpellier Computer Conference 1990 - Histoire et Informatique*, 1992, 623-636.

[37] M. LANA, *L'attribuzione di testi Gramsciani e i metodi quantitativi*, preprint, 2010.

[38] A. LEMPEL and J. ZIV, On the complexity of finite sequences, *IEEE Trans. Inform. Theory*, **IT-22** (1976), no. 1, 75-81.

[39] M. LI et al., An information-based sequence distance and its application to whole mitochondrial genome phylogeny, *Bioinformatics*, **17** (2001), no. 1, 149-154.

[40] *Literary and Linguistic Computing*, edited by M. Deegan, **13** (1998), no. 3.

[41] D. LOEWENSTERN et al., *DNA sequence classification using compression-based induction*, DIMACS Technical Report no. 95-04, 1995.

[42] A.A. MARKOV, Primer statisticheskogo issledovanija nad tekstom 'Evgenija Onegina' illjustrirujuschij svjaz' ispytanij v tsep (An example of statistical study on the text of'Eugene Onegin' illustrating the linking of events to a chain), *Izvestija Imp. Akademii nauk*, **VI** (1913), 153 -162.

[43] A.A. MARKOV, Ob odnom primeneni statisticheskogo metoda. (On some application of statistical method), *Izvestija Imp. Akademii nauk*, **VI** (1916), no. 4, 239-242.

[44] A.P. MASUCCI and G.J. RODGERS, Network properties of written human language, *Physical Review E*, **74** (2006), no.22, 026102.

[45] A.P. MASUCCI and G.J. RODGERS, Differences between normal and shuffled texts: structural properties of weighted networks, *Advances in Complex Systems*, **12** (2009), no. 1, 113-129.

[46] H. MAURER, F. KAPPE and B. ZAKA, Plagiarism - a survey, *Journal of Universal Computer Science*, **12** (2006), no. 8, 1050-1084.

[47] J. MEDORI, E. ATWELL, P. GENT and C. SOUTER, Customising a copying-identifier for biomedical science student reports: comparing simple and smart Analyses, in *Artificial Intelligence and Cognitive Science, 13th Irish Conference, AICS 2002, Limerick, Ireland, September 2002 (M. O'Neill et al., eds.)*, LNAI, **2464**, Springer-Verlag, Berlin, 2002, 228-233.

[48] T.C. MENDENHALL, The characteristic curves of composition, *Science*, **IX** (1887), 237-249.

[49] M.A. MONTEMURRO and D.H. ZANETTE, Entropic analysis of the role of words in literary texts, *Advances in Complex Systems*, **5** (2002), 7-17.

[50] M.A. MONTEMURRO and D.H. ZANETTE, Towards the quantification of the semantic information encoded in written language, *Advances in Complex Systems*, **13** (2010), no. 2, 135-153.

[51] D.S. ORNSTEIN and B. WEISS, Entropy and data compression schemes, *IEEE Transactions on Information Theory*, **39** (1993), no. 1, 78-83.

[52] H.H. OTU and K. SAYOOD, A new sequence distance measure for phylogenetic tree construction, *Bioinformatics* **19** (2003), no. 16, 2122-2130.

[53] J. PATRICK, Y. WANG and P. BUDD, An automated system for conversion of clinical notes into SNOMED Clinical Terminology, in *Proc. Australasian Workshop on Health Knowledge Management and Discovery (HKMD 2007), Ballarat, Australia (J.F. Roddick and J.R. Warren, eds.)*, CRPIT, **68**, ACS, 219-226.

[54] M. POTTHAST, B. STEIN, A. EISELT, A. BARRÓN-CEDEÑO and P. ROSSO, Overview of the 1st International Competition on Plagiarism Detection, in *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09) (B. Stein, P. Rosso, E. Stamatatos, M. Koppel and E. Agirre, eds.)*, CEUR-WS.org, 2009, 1-9.

[55] M. POTTHAST, B. STEIN, A. EISELT, A. BARRÓN-CEDEÑO and P. ROSSO, Overview of the 2nd International Competition on Plagiarism Detection, in publication in *CLEF 2010, Conference on Multilingual and Multimodal Information Access Evaluation, 20-23 September 2010, Padua*, 2010.

[56] A. PUGLISI, D. BENEDETTO, E. CAGLIOTI, V. LORETO and A. VULPIANI, Data compression and learning in time sequences analysis, *Physica D*, **180** (2003), no. 1-2, 92-107.

[57] B. SAUSSOL, Recurrence rate in rapidly mixing dynamical systems, *Discrete and Continuous Dynamical Systems A*, **15** (2006), 259-267.

[58] L. SEAWARD and S. MATWIN, Intrinsic plagiarism detection using complexity analysis, in *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09) (B. Stein, P. Rosso, E. Stamatatos, M. Koppel and E. Agirre, eds.)*, CEUR-WS.org, 2009, 56-61.

[59] C.E. SHANNON, A mathematical theory of communication, *The Bell System Technical Journal*, **27** (1948), 379-423, 623-656.

[60] P.C. SHIELDS, *The ergodic theory of discrete sample paths*, Graduate Studies in Mathematics, vol. 13, American Mathematical Society, Providence, RI, 1996.

[61] E. STAMATATOS, Ensemble-based author identification using character n-grams, *Proceedings of the 3rd International Workshop on Text-Based Information Retrieval (TIR'06)*, 2006, 41-46.

[62] E. STAMATATOS, A survey of modern authorship attribution methods, *Journal of the American Society for Information Science and Technology*, **60** (2009), no. 3, 538-556.

[63] E. STAMATATOS, Intrinsic plagiarism detection using character $n-$gram profiles, in *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09) (B. Stein, P. Rosso, E. Stamatatos, M. Koppel and E. Agirre, eds.)*, CEUR-WS.org, 2009, 38-46.

[64] B. STEIN, S. MEYER ZU EISSEN and M. POTTHAST, Strategies for Retrieving Plagiarized Documents, in *30th Annual International ACM SIGIR Conference (Clarke, Fuhr, Kando, Kraaij, and de Vries, eds.)*, ACM, 2007, 825-826.

[65] W.J. TEAHAN, Text classification and segmentation using minimum cross-entropy, in *Proceedings of the International Conference on Content-based Multimedia Information Access (RIAO 2000)*, C.I.D.-C.A.S.I.S, Paris, 2000, 943-961.

[66] M. TRIBUS and E.C. McIRVINE, Energy and information, *Scientific American*, **224** (1971), 179-184.

[67] A.D. WYNER and J. ZIV, Fixed data base version of the Lempel-Ziv data compression algorithm, *IEEE Transactions on Information Theory*, **37** (1991), no. 3, 878-880.

[68] A.D. WYNER and J. ZIV, The sliding-window Lempel-Ziv algorithm is asymptotically optimal, *Proceedings of the IEEE*, **82** (1994), no. 6, 872-877.

[69] A.D. WYNER, J. ZIV and A.J. WYNER, On the role of pattern matching in information theory, *IEEE Transactions on information Theory*, **44** (1998), no. 6, 2045-2056.

[70] J. ZIV and A. LEMPEL, A universal algorithm for sequential data compression, *IEEE Transactions on Information Theory*, **23** (1977), no. 3, 337-343.

[71] J. ZIV and A. LEMPEL, Compression of individual sequences via variable-rate coding, *IEEE Transactions on Information Theory*, **IT-24** (1978), no. 5, 530-536.

[72] J. ZIV, Coding theorems for individual sequences, *IEEE Transactions on information Theory*, **IT-24** (1978), no. 4, 405-412.

[73] J. ZIV and N. MERHAV, A measure of relative entropy between individual sequences with application to universal classification, *IEEE Transactions on Information Theory*, **39** (1993), no. 4, 1270-1279.